Active Magnetic Anomaly Detection Using Multiple Micro Aerial Vehicles

Philip M. Dames[1], Mac Schwager[2], Daniela Rus[3], and Vijay Kumar[1]

***Abstract*—Magnetic Anomaly Detection (MAD) is an important problem in applications ranging from geological surveillance to military reconnaissance. MAD sensors detect local disturbances in the magnetic field, which can be used to detect the existence of and to estimate the position of buried, hidden, or submerged objects, such as ore deposits or mines. These sensors may experience false positive and false negative detections and, without prior knowledge of the targets, can only determine proximity to a target. The uncertainty in the sensors, coupled with a lack of knowledge of even the existence of targets, makes the estimation and control problems challenging. We utilize a hierarchical decomposition of the environment, coupled with an estimation algorithm based on random finite sets, to determine the number of and the locations of targets in the environment. The small team of robots follow the gradient of mutual information between the estimated set of targets and the future measurements, locally maximizing the rate of information gain. We present experimental results of a team of quadrotor micro aerial vehicles discovering and localizing an unknown number of permanent magnets.**

***Index Terms*—Networked Robots; Probability and Statistical Methods; Reactive and Sensor-Based Planning**

I. INTRODUCTION

MAGNETIC Anomaly Detection (MAD) has applications in military, humanitarian, and industrial settings. MAD sensors detect disturbances in the Earth's magnetic field that indicate the presence of nearby objects such as land and sea mines [1], ore and mineral deposits [2], and vehicles [3]. In all of these applications, the task is to determine if there are any objects of interest within the search area and, if any targets exist, the number of and locations of the targets. The search areas may span large geographic distances and the tasks may be very dangerous. Using a small team of robots to autonomously gather information about the targets keeps humans from performing these dull, dirty, and dangerous tasks.

Manuscript received: August 31, 2015; Revised November 13, 2015; Accepted December 15, 2015.

This paper was recommended for publication by Editor-in-Chief Antonio Bicchi and Editor Nak Young Chong upon evaluation of the Associate Editor and Reviewers' comments. *This work was funded in part by ONR MURI Grants N00014-07-1-0829, N00014-09-1-1051, and N00014-09-1-1031, ARO Grant W911NF-13-1-0350, NSF Grant IIS-1426840, and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. P. Dames was supported by the Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

[1]P. Dames and V. Kumar are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA {pdames, kumar}@seas.upenn.edu

[2]M. Schwager is with the Department of Aeronautics and Astronautic, Stanford University, Stanford, CA 94305, USA schwager@stanford.edu

[3]D. Rus is with the Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT, Cambridge, MA 02139, USA rus@csail.mit.edu

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

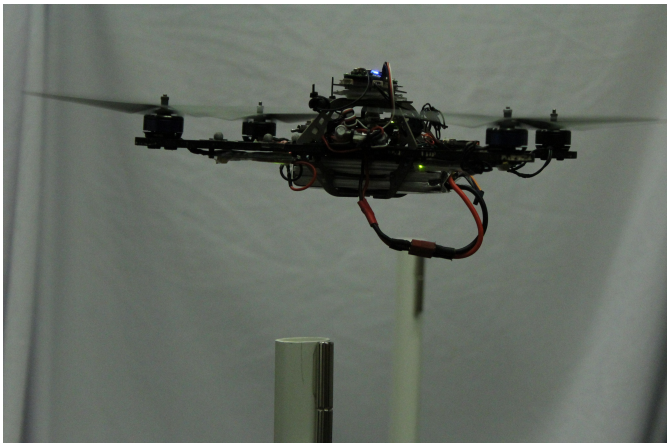


Fig. 1. Photo of an Ascending Technology Hummingbird MAV hovering over a magnetic target. A second target may be seen in the background.

In this paper we propose a unified estimation and control strategy for a small team of robots to explore an environment in search of an unknown number of targets. MAD sensors often have a low signal-to-noise ratio, making target detection difficult [3]. One of the most common methods for MAD is based on orthogonal basis functions [4]. Instead, we utilize an estimation algorithm that explicitly models the possibility of false negative and false positive detections. This algorithm is based on the mathematical concept of a Random Finite Set (RFS) [5]. An RFS is a random variable with realizations that are finite sets, jointly modeling the number of the targets as the size of the set and the state of each target as the individual elements of the set. Estimation algorithms based on RFSs are becoming increasingly popular and have been used in applications such as vehicle tracking [6], Simultaneous Localization and Mapping (SLAM) [7], robot localization [8], and target search [9]–[11].

The robot team uses the resulting estimate of the target set to make control decisions. Robots follow the gradient of mutual information between the target set and the future measurements, locally maximizing the expected gain in information. This approach, known as "information surfing," has been used in a variety of source seeking applications [9], [10], [12]–[14]. More generally, information-based control laws have been used to autonomously search for targets using a team of mobile sensors [6], [11], [15], [16].

While we focus on applications of MAD sensors in this paper, the estimation and control framework work for any sensor that provides detection data. The work in this paper builds on our previous work [9], [10], [17]. In [9] we initially proposed

the estimation and control strategies and also considered the presence of environmental hazards. In [10] we developed a decentralized version of the estimation and control strategies. In [17] we applied the estimation and control framework from [9] to a single ground robot equipped with a monocular camera to detect, localize, and collect multiple objects. Compared to [17], the sensors used in this work are much coarser, providing only a single bit of data. This paper has three main contributions over our previous work. One, unlike our previous work, which was based on simulated experiments or a single ground robot in hardware, we present a series of experiments using Ascending Technology Hummingbird quadrotor micro aerial vehicles (MAVs) equipped with magnetometer sensors. Magnetometers measure the strength of the local magnetic field and, in this situation, are used to detect anomalies in order to localize targets. We use two different single-robot systems in the hardware experiments and teams of 1–4 heterogeneous robots in simulation. Two, we experimentally characterize the detection statistics of the sensors, a critical component to be able to validate the effectiveness of our estimation and control strategy. Three, we prove that the control computations depend only on the visible subset of the environment, greatly improving the speed of the computations.

II. PROBLEM FORMULATION

The objective of the mission is to identify the correct target set, which includes both target count and target positions, as quickly as possible. Let $\mathbf{q}$ denote the pose of an MAV within the environment. Let $\mathbf{x}$ denote the pose of an individual target and $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote a set of targets.

A. Sensors

Each MAV is equipped with a sensor that is able to detect the presence of nearby targets within the sensor field of view (FoV). Let $p_d(\mathbf{x} \mid \mathbf{q})$ be the probability of a sensor with pose $\mathbf{q}$ detecting a single target with pose $\mathbf{x}$, where p_d is 0 for all targets outside of the FoV. Note that this sensor is *extremely* coarse, providing only a single bit of information about whether there are any targets within the FoV. Such coarse sensors experience false negative detections, with probability $1 - p_d(\mathbf{x} \mid \mathbf{q})$, and may also experience false positive detections, with probability p_{fp}.

For multiple targets, the probability of detection is

$$p_d(X \mid \mathbf{q}) = 1 - (1 - p_{\text{fp}}) \prod_{\mathbf{x} \in X} \big(1 - p_d(\mathbf{x} \mid \mathbf{q})\big), \quad (1)$$

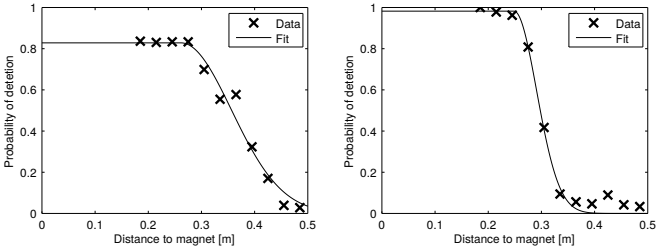
since the only way to *not* get a detection is to have *no* false positive detections and to *not* detect any of the targets. The measurements are binary, with $z = 1$ corresponding to the sensor detecting the presence of at least one target within the FoV. The measurement model is

$$g(z \mid X, \mathbf{q}) = \begin{cases} p_d(X \mid \mathbf{q}), & z = 1 \\ 1 - p_d(X \mid \mathbf{q}), & z = 0. \end{cases} \quad (2)$$

The sensors that we utilize in this work are the magnetometers found onboard the MAVs. These sensors are much smaller and lower cost than the traditional MAD sensors mounted to full-scale aircraft or boats to detect ore deposits or submarines and, as a result, are much coarser. Additionally, it is well known that magnetometers are not reliable indoors due to the presence of metal building materials, electrical wiring, and other such components. This problem is further exacerbated by the robots themselves: the drive motors contain permanent magnets, the wires to the drive motors have very high current and the magnetometers are located near the onboard computers and wireless antennae. Furthermore, the strength and direction of a magnetic field depends highly on the orientation of the magnetic source. These factors all make the inference problem difficult.

TABLE I
BEST FIT MAD SENSOR PARAMETERS.

MAV	p_{fn}	R_0 [m]	σ_R [m]	R_1 [m]	p_{fp}	ϵ_m
Kilo	0.172	0.262	0.0948	0.5	0.00320	0.55
Papa	0.0177	0.249	0.0425	0.5	0.0138	0.03



(a) MAV Kilo (b) MAV Papa

Fig. 2. Experimentally determined MAD sensor detection models used for target detection and localization.

To account for all of these uncertainties we model the magnetometer as a binary sensor, returning a positive measurement if the magnetic field is "sufficiently" disturbed from the nominal value. Letting $r = |\mathbf{x} - \mathbf{q}|$, the probability of detection takes the form

$$p_d(\mathbf{x} \mid \mathbf{q}) = \begin{cases} 1 - p_{\text{fn}} & r < R_0, \\ (1 - p_{\text{fn}}) \exp\left(-\frac{(r - R_0)^2}{2\sigma_r^2}\right) & R_0 \le r \le R_1, \\ 0 & R_1 < r. \end{cases} \quad (3)$$

Here, p_{fn} is the probability of a false negative, R_0 is the radius inside which the probability of detection is constant, σ_r is the rate at which the probability of detection drops off with distance, and R_1 is the maximum detection range of the sensor. The values of the parameters depend on the specific sensor and robot being used. Fig. 2 shows the experimentally derived sensor models for the two MAV platforms we use, Kilo and Papa. Table I lists the best fit parameters for the two MAV platforms. See Section III-A for further details on the sensor model and characterization.

Note that the detection model, $p_d(\mathbf{x} \mid \mathbf{q})$, has a finite FoV, since any real-world sensor will not be able to see the entire environment. Thus, target sets that differ only *outside* of the FoV of a sensor appear identical. We can leverage this fact to improve the computational efficiency of the estimation and control algorithms, but we must first define a projection opera-

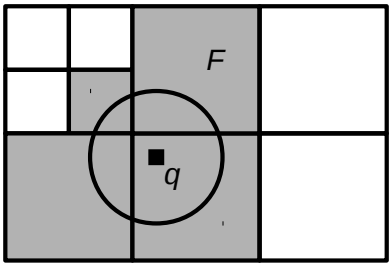


Fig. 3. An example environment with one robot with pose q and a circular sensor FoV. The cells within the FoV, F, are highlighted in gray.

tor. Let F be the set of cells that have a non-empty intersection with the sensor FoV. See Fig. 3 for a simple example of this. Consider the projection of a target set $r(X) = X \cap F$ and let $V = r(X)$ denote a target set projected onto the sensor FoV. Note that this map is surjective but not injective as long as F is a proper subset of the environment, so no inverse mapping exists. The right inverse still exists, where $r((r)^{-1}(V)) = V$ but $(r)^{-1}(r(X)) \neq X$. The right inverse of the projection is $(r)^{-1}(V) = \{X \mid r(X) = V\}$, which in general returns multiple values that correspond to all of the sets that appear identical to a sensor with FoV F.

Lemma 1: The measurement model is unaffected by the project operator, *i.e.,* $g(z \mid X, \mathbf{q}) = g(z \mid r(X), \mathbf{q})\ \forall X$.

Proof: By definition, $p_d(\mathbf{x} \mid \mathbf{q}) = 0$ for all $\mathbf{x} \notin F$. From (1), we see that

$$\begin{aligned} p_d(X \mid \mathbf{q}) &= 1 - (1 - p_{\text{fp}}) \prod_{\mathbf{x} \in X} \big(1 - p_d(\mathbf{x} \mid \mathbf{q})\big) \\ &= 1 - (1 - p_{\text{fp}}) \prod_{\mathbf{x} \in r(X)} \big(1 - p_d(\mathbf{x} \mid \mathbf{q})\big) \prod_{\mathbf{x} \notin r(X)} 1 \\ &= p_d(r(X) \mid \mathbf{q}). \end{aligned} \tag{4}$$

Thus, $g(z \mid X, \mathbf{q}) = g(z \mid r(X), \mathbf{q})$. ■

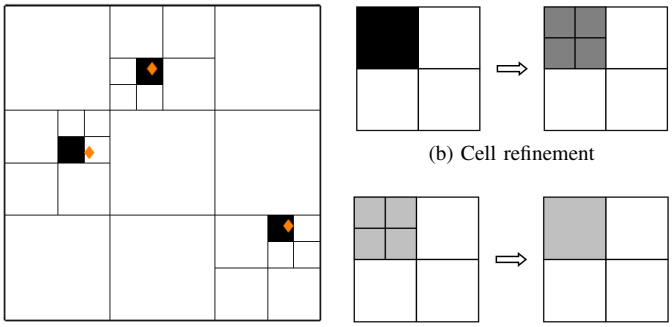
B. Estimation

As robots explore the environment, they exchange their measurements and poses. The robots then feed this data into a recursive Bayesian filter to estimate the target locations from the collected data. However, unlike many multi-target estimation algorithms such as the Probability Hypothesis Density (PHD) filter [5] or occupancy grids [18], we cannot assume that targets are independent due to the coarse nature of the sensor. Instead we maintain a distribution over sets, as we did in our previous work [9], [10]. We keep the number of RFSs at a computationally tractable level by using a hierarchical decomposition of the environment rather than a fixed grid, as Fig. 4a shows. This is similar in spirit to the approach in Ristic and Vo [11], but we enumerate all possible sets over a hierarchical decomposition instead of using a randomly sampled collection of target sets.

The update equations for the Bayesian filter are

$$p_t(X \mid z) = \frac{g(z \mid X, \mathbf{q}) p_{t-1}(X)}{\sum_{X'} g(z \mid X', \mathbf{q}) p_{t-1}(X')}, \tag{5}$$

where $z \in \{0, 1\}$ are binary measurements. Note that in the estimation equations we overload the variable X to be a set



(a) Example target set (b) Cell refinement (c) Cell merge

Fig. 4. (a) An example realization of a RFS over the hierarchical grid. The black cells are the estimated occupied cells and the white cells are empty. This is close to the true set, shown by the orange diamonds. (b) A cell refinement procedure, where a large occupied cell is divided into four smaller cells with uniform occupancy probability. The shading in each cell indicates the probability that that cell is occupied, with white being 0 and black being 1. (c) A grid merging procedure, where four empty sub-cells with the same parent cell are merged to form the parent cell.

Algorithm 1 Add cell c

$\mathcal{X}' \leftarrow \mathcal{X}$ ▷ Copy existing sets
$C \leftarrow C + 1$ ▷ Add one to cell count
for $X \in \mathcal{X} \mid |X| < N$ **do**
  $p'(X) \leftarrow \frac{1}{2} p(X)$
  $p'(\{X \cup \{c\}\}) \leftarrow \frac{1}{2} p(X)$
  $\mathcal{X}' \leftarrow \mathcal{X}' \cup \{X \cup \{c\}\}$ ▷ Add set with new cell
end for
$\mathcal{X} \leftarrow \mathcal{X}'$

of cells, $X = \{c_1, \ldots, c_n\}$, which indicates that there are n targets in n distinct cells (*i.e.,* $\mathbf{x}_i \in c_i$, $i = 1, \ldots, n$). As can be seen, the complexity of the filter updates depends upon the number of sets. If the number of potential targets was equal to the number of cells (C), there would be 2^C possible sets. However, if we limit the number of targets (N) to be small compared to C, then the number of sets decreases to $\mathcal{O}(C^N)$. To further reduce the complexity, we utilize a hierarchical decomposition of the environment, here based on a quadtree, to limit the number of cells, C. When the probability that a cell is occupied is higher than a pre-specified threshold, the cell is subdivided into a collection of smaller, disjoint sets, as Fig. 4b shows. For a quadtree, this process involves removing the parent cell and then adding four child cells. Similarly, when the probability that all of the child cells of a parent cell are empty is higher than a desired threshold, the cells are merged back into the parent cell, as Fig. 4c shows. Algorithms 1 and 2 outline the cell addition and removal procedures, respectively.

Algorithm 2 Remove cell c

$\mathcal{X}' \leftarrow \mathcal{X}$ ▷ Copy existing sets
$C \leftarrow C - 1$ ▷ Remove one from cell count
for $X \in \mathcal{X} \mid c \in X$ **do**
  $p(Y) \leftarrow \sum_{X \mid X \setminus \{c\} = Y} p(X)$
  $\mathcal{X}' \leftarrow \mathcal{X}' \setminus \{X\}$ ▷ Remove all sets with c
end for
$\mathcal{X} \leftarrow \mathcal{X}'$

We can leverage the projection operation, $r(X)$, to improve the efficiency of the filter updates by only performing the update over the collection of projected sets.

Theorem 1: The filter update in (5) is equivalent to

$$p_t(X \mid z) = \frac{p_t(r(X) \mid z)}{p_{t-1}(r(X))} p_{t-1}(X). \tag{6}$$

Proof: Consider the collection of target sets containing targets in F, $\mathcal{V} = \cup_{X \in \mathcal{X}} (X \cap F)$. Using Bayes' theorem, for any $V \in \mathcal{V}$ we have

$$p_t(V \mid z, \mathbf{q}) = \frac{g(z \mid V, \mathbf{q}) p_{t-1}(V)}{p(z \mid \mathbf{q})}. \tag{7}$$

Using (5), (7), and Lemma 1, we see that

$$\frac{p_t(X \mid z)}{p_{t-1}(X)} = \frac{g(z \mid X, \mathbf{q})}{p(z \mid \mathbf{q})} = \frac{g(z \mid r(X), \mathbf{q})}{p(z \mid \mathbf{q})} = \frac{p_t(r(X) \mid z)}{p_{t-1}(r(X))}.$$

Rearranging terms leads to the desired result. ∎

Note that with an initially coarse discretization of the environment, some cells may be large compared to the sensor FoV. Since targets may be located anywhere within the cell, we must take into account the partial visibility of some cells. Given this, the probability of detection for cell c is given by

$$p_d(c \mid \mathbf{q}) = \frac{1}{|c|} \int_{\mathbf{x} \in c} p_d(\mathbf{x} \mid \mathbf{q})\, d\mathbf{x} \approx \frac{1}{m} \sum_{k=1}^{m} p_d(\mathbf{x}_{c,k} \mid \mathbf{q}), \tag{8}$$

where $|c|$ is the area of the cell c. This integral is approximated by a sum over a set of m points in the cell, $\{\mathbf{x}_{c,k}\}_{k=1}^{m} \in c$, which, given no available information beyond our binary detections, are distributed uniformly over the cell. The simplest approach is to use the cell centroids. However, multiple points should be used for cells that are large compared to the sensor field of view so that at least one such point is always within the FoV.

To implement Algorithms 1 and 2, the collection of sets over a cellular representation of the environment can be efficiently encoded using the binary representation of integers. Adding a cell is done by allowing the next highest bit to be set, and removing a cell is done by shifting all of the bits above the cell to be removed. The number of targets in a set is the number of bits set to high. The rest is a matter of bookkeeping, making sure each cell is associated to a unique bit in the integer.

C. Control

The robots utilize the output of the filter, combined with a sensor model and knowledge of each others' poses, to make control decisions. Specifically, the robots follow the gradient of the mutual information between the set of targets and the possible future measurements. This leads the robots to locally optimize the information gain, a strategy known as "information surfing" [13]. While planning over a longer time horizon may be preferable in some scenarios [6], [19]–[21], the limited information provided by a binary sensor would require such a long horizon as to be computationally infeasible in practice.

Mutual information was first defined by Shannon [22],

$$I[\mathcal{X}; \mathcal{Z} \mid \mathbf{q}] = \sum_{X \in \mathcal{X}} \sum_{\mathbf{z} \in \mathcal{Z}} p(\mathbf{z}, X \mid \mathbf{q}) \log \frac{p(\mathbf{z}, X \mid \mathbf{q})}{p(\mathbf{z} \mid \mathbf{q}) p(X)}. \tag{9}$$

This quantifies the amount of dependence between random variables, in this case the target set X (of cell labels) and the vector of future measurements of all of the robots in the team $\mathbf{z}$. By following the gradient of mutual information with respect to the pose of the robots, we expect the team to locally maximize the rate of information gain.

We have previously shown in [9] that the gradient of mutual information with respect to the pose of robot r is

$$\frac{\partial I[\mathcal{X}; \mathcal{Z} \mid Q]}{\partial \mathbf{q}^r} = \sum_{\mathbf{z} \in \mathcal{Z}} \sum_{X \in \mathcal{X}} \frac{\partial g(\mathbf{z} \mid X, Q)}{\partial \mathbf{q}^r} p_t(X) \times \log \frac{g(\mathbf{z} \mid X, Q)}{\sum_{X \in \mathcal{X}} g(\mathbf{z} \mid X, Q) p_t(X)}, \tag{10}$$

where $Q = \{\mathbf{q}^1, \ldots, \mathbf{q}^R\}$. Here $p_t(X)$ comes from (5) and $g(\mathbf{z} \mid X, Q) = \prod_{r=1}^{R} g(z^r \mid X, \mathbf{q}^r)$, since measurements from different sensors are conditionally independent give the target set. The gradient of this is

$$\frac{\partial g(\mathbf{z} \mid X, Q)}{\partial \mathbf{q}^r} = \frac{g(\mathbf{z} \mid X, Q)}{g(z^r \mid X, \mathbf{q}^r)} \frac{\partial g(z^r \mid X, \mathbf{q}^r)}{\partial \mathbf{q}^r} \tag{11}$$

and the gradient of (1) is

$$\frac{\partial g(z \mid X, \mathbf{q})}{\partial \mathbf{q}} = (-1)^{1-z} g(z = 0 \mid X, \mathbf{q}) \times \sum_{x \in X} \frac{1}{1 - p_d(x \mid \mathbf{q})} \frac{\partial p_d(x \mid \mathbf{q})}{\partial \mathbf{q}}. \tag{12}$$

We also take advantage of the projection operator to greatly improve the efficiency of the control computations, replacing all instances of X by $r(X)$.

Theorem 2: Mutual information is not affected by the projection operator, *i.e.,* $I[\mathcal{X}; \mathcal{Z} \mid \mathbf{q}] = I[\mathcal{V}; \mathcal{Z} \mid \mathbf{q}]$.

Proof: Using (9) and Lemma 1,

$$\begin{aligned} I[\mathcal{X}; \mathcal{Z} \mid \mathbf{q}] &= \sum_{X \in \mathcal{X}} \sum_{\mathbf{z} \in \mathcal{Z}} g(\mathbf{z} \mid X, \mathbf{q}) p(X) \log \frac{g(\mathbf{z} \mid X, \mathbf{q})}{p(\mathbf{z})} \\ &= \sum_{V \in V} \sum_{\mathbf{z} \in \mathcal{Z}} g(\mathbf{z} \mid V, \mathbf{q}) \log \frac{g(\mathbf{z} \mid V, \mathbf{q})}{p(\mathbf{z})} \\ &\quad \times \underbrace{\sum_{X \in \mathcal{X} \mid r(X) = V} p(X)}_{= p(V)} \\ &= \sum_{V \in V} \sum_{\mathbf{z} \in \mathcal{Z}} g(\mathbf{z} \mid V, \mathbf{q}) p(V) \log \frac{g(\mathbf{z} \mid V, \mathbf{q})}{p(\mathbf{z})} \\ &= I[\mathcal{V}; \mathcal{Z} \mid \mathbf{q}]. \end{aligned}$$

∎

Corollary 1: The gradient of mutual information is not changed by the project operator.

Remark 1: For multiple robots the projection becomes $r(X) = X \cap (\cup_r F^r)$.

The robots follow the gradient of mutual information with respect to their poses,

$$\mathbf{q}_{t+1}^r = \mathbf{q}_t^r + k \frac{\frac{\partial I[\mathcal{V}; \mathcal{Z} \mid Q]}{\partial \mathbf{q}^r}}{\left| \frac{\partial I[\mathcal{V}; \mathcal{Z} \mid Q]}{\partial \mathbf{q}^r} \right| + \epsilon}, \tag{13}$$

where k is the step size and $\epsilon = 10^{-20} \ll 1$ prevents numerical errors when the gradient is zero. The computational

complexity of this is $\mathcal{O}(2^R C_F^N)$, where N is the maximum number of targets, R is the number of robots, and C_F is the number of cells within the footprint of the robot team. In [10] we showed that the control actions are independent for robots that do not observe any of the same cells, *i.e.,* $F^i \cap F^j = \varnothing$ for $i \neq j$, decreasing both R and C_F. When the information gain is extremely low, *i.e.,* $I[\mathcal{V}; \mathcal{Z} \mid Q] < 10^{-6}$, the robot instead drives towards the cell with the highest uncertainty, *i.e.,* the cell with probability of occupancy closest to 0.5.

The collision avoidance algorithm used in the multi-robot experiments is myopic, with a robot backing up, *i.e.,* following the negative gradient, if it would come into collision with another robot. This occasionally causes the robots to step forwards then backwards repeatedly. When this occurs while a target is located near the edge of the sensor footprint it makes target localization more difficult and unreliable since the detection likelihood is lowest at the edge of the footprint. This would be less of an issue for robots with sensor footprints that are significantly larger than the physical footprint of the robot. In these experiments, the sensor has a radius of 0.5 m and the robot has a radius of 0.375 m.

III. EXPERIMENTAL RESULTS

MAD sensors for large-scale surveillance are finely calibrated to detect very subtle disturbances in the Earth's magnetic field. The noisy, scaled-down laboratory environment is representative of more complex, real-world environments.

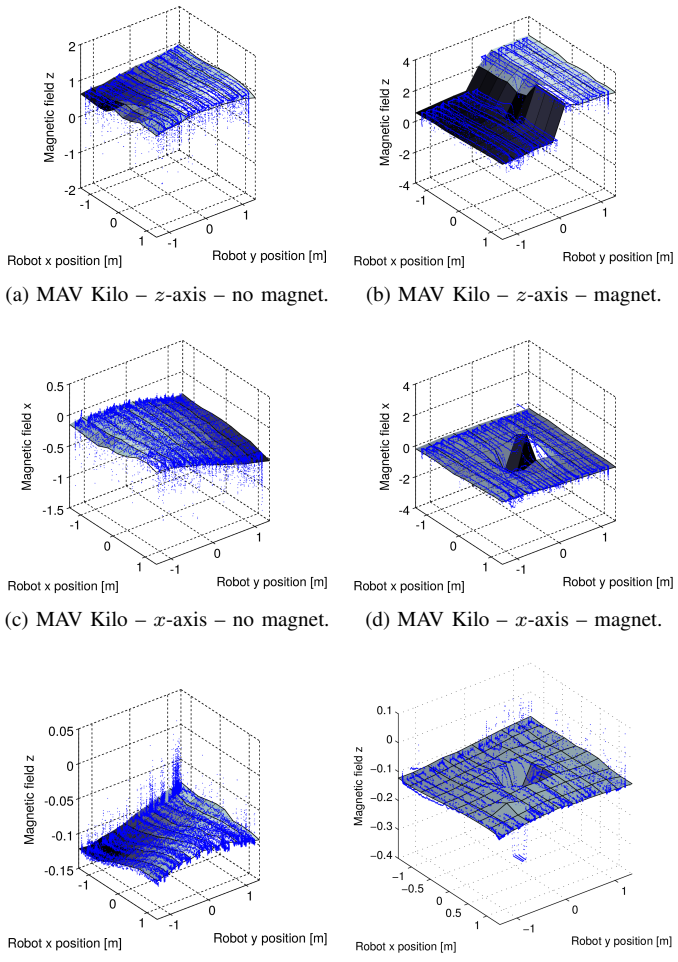
A. Sensor Characterization

We first ran experiments to characterize the detection statistics for the 3-axis magnetometer on board the AscTec Hummingbird MAV platform, shown in Fig. 1. Each target consists of two cylindrical neodymium magnets[1] with the axis of the magnets aligned with the z-axis of the global coordinate frame and having a combined length of 10 cm.

To learn the baseline magnetic field, we flew the Hummingbird through a lawnmower pattern over a 2.6×2.6 m area at a constant height of approximately 1.3 m. Fig. 5 shows that the magnetic field experiences significant changes over the area covered by the robot, varying by almost an order of magnitude. This is likely due to the presence of a metal staircase and other large objects in addition to building materials, electrical wires, and other robots in the laboratory space. Despite the large variance in measurements taken at the same location, the average field changes smoothly over the environment. We fit a nominal field strength by dividing the area into grid cells with size 30 cm and taking the empirical mean of all of the magnetic field readings taken within that cell. Angermann et al. [23] create a similar map of the ambient indoor magnetic field, using the local structure in order to localize robots within a workspace.

To characterize the detection statistics, we flew the same pattern with a single magnetic target positioned at $(0, 0, 1.12)^T$ m. The resulting magnetic field readings are

[1]K&J Magnetics, Inc. D8Y0, https://www.kjmagnetics.com/proddetail.asp?prod=D8Y0a



(a) MAV Kilo – z-axis – no magnet. (b) MAV Kilo – z-axis – magnet.

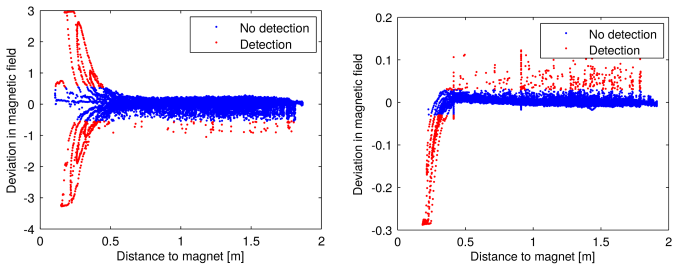
(c) MAV Kilo – x-axis – no magnet. (d) MAV Kilo – x-axis – magnet.

(e) MAV Papa – z-axis – no magnet. (f) MAV Papa – z-axis – magnet.

Fig. 5. Experimental results of the magnetic field strength as a function of the 2D position of the MAVs in the baseline training runs. Blue dots indicate the individual data points and the grayscale surface shows the average value in each cell. The magnetic field strength along the z-axis for MAV Kilo is shown (a) before and (b) after a magnet is placed at (0, 0) m. The magnetic field strength along the x-axis for MAV Kilo is shown (c) before and (d) after a magnet is placed at (0, 0) m. The magnetic field strength along the z-axis for MAV Papa is shown (e) before and (f) after a magnet is placed at (0, 0) m.

plotted as a function of the distance to the target in Fig. 6. We use the deviation from the nominal fields shown in Fig. 5c and Fig. 5e to determine the presence of a target. From Fig. 6 it is evident that far away from the target the deviations from the nominal field are relatively small, though with a few clutter detections. Near to the targets there are significant deviations in the magnetic field, with an approximate detection radius of 0.5 m. We have colored the data points according to whether they are inliers or outliers, using a threshold ϵ_m on the deviation in the field strength. We selected ϵ_m values (in Table I) such that increasing the value results in significantly more false positive detections and a small decrease in the rate of false negative detections.

Note in Fig. 5b that for MAV Kilo there is a step increase induced in the readings along the z-axis of the magnetometer after the robot passes directly over the magnet. This does not occur for MAV Papa. The phenomenon was repeatable in the training runs, but presents problems for actual experimental



(a) MAV Kilo. (b) MAV Papa

Fig. 6. Experimental results of the deviation of the magnetic field due to the addition of a magnet as a function of the true distance to the magnet for (a) MAV Kilo and (b) MAV Papa.

trials as the robot does not know the locations of the magnets *a priori*. To avoid this problem, we instead use the magnetometer along the x-axis for MAV Kilo. However, the deviation along the x-axis, unlike the z-axis, is not isotropic and changes signs depending on what side of the magnet the robot is on, as Fig. 5d shows. In order to keep the models for the two robots similar, we ignore this change in the sign of the deviation for MAV Kilo and only consider the absolute value of the deviation in the magnetic field.

We use this data to characterize the detection and clutter models for the MAD sensor. The probability of a false positive is computed using the ratio of detections to all measurements outside of the sensing radius. To compute the detection statistics, we divide the distance from the robot to the target into bins (of width 3 cm) and look at the detection rate within each bin, using the thresholds from above to determine true versus missed detections. Fig. 2 shows this experimental data and the best fit detection models. For both robots, the detection rate is relatively high and constant when the distance is small and falls sharply towards zero as the distance increases. Given this, and letting $r = |\mathbf{x} - \mathbf{q}|$, we model the probability of detection in (3). To find the best fit parameters, given in Table I, we perform a brute-force search over a range of the parameter space, selecting the model with the minimum sum-of-squares error between the data points the model. p_{fp} is the probability of a false positive detection and is found by counting the empirical fraction of detections when the magnet was further than the maximum sensing range. The differences in the models, particular the false negative rate, is due to the different magnetometer axes being used by the robots.

B. Single Robot Results

We conduct a series of hardware and simulation experiments with a single robot to test the performance of the search algorithm in the MAD setting and to validate the performance of the simulation environment. With each of the two robots, Kilo and Papa, we performed three individual hardware trials and five simulation trials. The hardware experiments are performed in a Vicon motion capture system, which provides each robot with an accurate estimate of its pose. The robots explore a 2×2 m area that is divided into cells with a maximum edge length of 50 cm and a minimum length of 12.5 cm. The minimum cell size is on the same length scale as the 10 cm long targets and significantly smaller than the 1 m diameter sensor footprint.

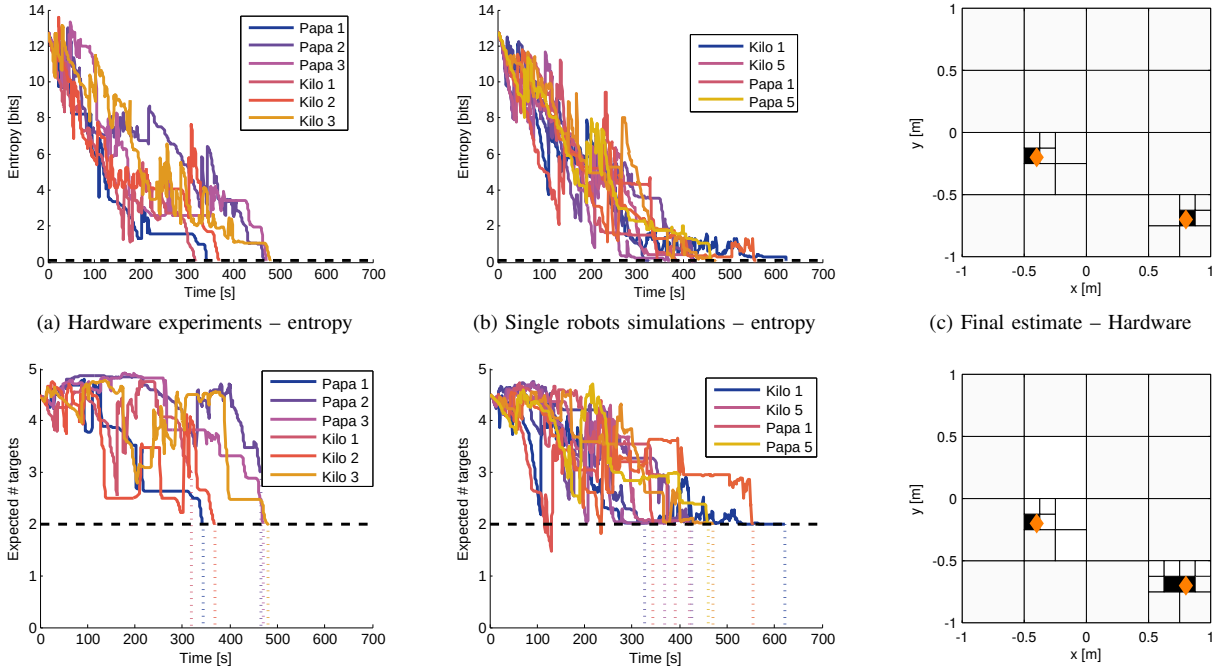
Fig. 7 shows the resulting evolution of the entropy of the target set and of the expected number of targets over time. Note that the time includes both computation and movement. The overall behavior is similar across both hardware and simulation experiments. The target entropy decreases quickly at first and has several step increases as cells are subdivided before finally reaching the desired level of 0.1 bits. Similarly, the expected number of targets begins near 4.5 before reaching a final value near 2.0, the true value. For both the simulated and hardware experiments, there was a single trial where the final expected number of targets was 3.0. Fig. 7f shows the final estimate for one such occurrence, where the robot incorrectly determined that two adjacent cells both contain a target. There was also a single trial in both the hardware and simulated systems where one of the targets was mis-localized, with the true target being in a cell adjacent to the final estimated position. Fig. 8 shows the statistics of the time to completion. The minimum and median times are very similar, at 320 s and 417.5 s for the hardware experiments and 326 s and 422.5 s for the simulation experiments.

C. Multi-Robot Results

We use the simulation environment to test the performance with teams of 2, 3, and 4 MAVs, avoiding complex, unmodeled interactions between physical robots, such as the magnetic field induced by the motors of one robot interfering with the magnetometer readings on the other robots. Given the level of similarity between the performance of the MAD system in the previous hardware and simulation experiments, we feel confident that the simulation results could be replicated in hardware. The sensor models used in these trials match those of the previous experiments, with at least one MAV matching Kilo and at least one matching Papa so that the team had heterogeneous sensor models.

Fig. 8 shows the statistics of the time to completion for the 2 target scenario from the single-robot trials. Here we see that teams of 2 and 3 robots are able to complete the task more quickly than a single robot. However, a 4 robot team takes longer and only completes the task within the 700 s time budget in 5 of the 10 trials. This is due to the added computational complexity of planning for four robots. Table II shows this more clearly, with 4 robots spending 48.2% of the total time computing instead of moving, a significantly higher percentage than smaller teams.

In addition to the 2 target scenario that we test in hardware, we test the system with 0, 1, and 4 targets. Fig. 9 shows the statistics in the time to completion. One trend that quickly emerges is that it takes longer to localize more targets. This is due to the extreme coarseness of the sensor, providing only a single bit of data with each measurement. The accompanying video shows how robots must circle around targets, taking many measurements, in order to localize the targets. Since this must be done for each target, increasing the number of targets significantly increases the time to completion. Also, adding more targets increases the total number of cells necessary to



(a) Hardware experiments – entropy

(b) Single robots simulations – entropy

(c) Final estimate – Hardware

(d) Hardware experiments – expected number of targets

(e) Single robots simulations – expected number of targets

(f) Final estimate – Hardware

Fig. 7. Experimental results. (a–b) show the time evolution of the entropy of the target estimate and (d–e) show the time evolution of the expected number of targets for a single robot in hardware and in simulation, respectively. (c) and (f) show localization results for two runs of a single real-world quadrotor. The orange diamonds indicate the true target positions and shading within each cell is the probability of occupancy.

localize all of the targets. This increases the computational load since the estimation and control algorithms scale polynomially in the number of cells.

The other trend that emerges from this data is that there is a clear tradeoff between the increased sensing provided by adding more robots and the increased computational burden of planning for a larger team. At a certain point the search area becomes saturated with robots: the sensor footprint of each robot covers 19.6% of the search area and each robot covers 11.0% of the search area. Table II shows the effects of this tradeoff, with larger teams spending significantly more time computing and less time moving than small teams. This is particularly true with a larger number of targets, as Fig. 8, Fig. 9c, and Table III show. In the 4 target case, the team is almost never able to complete the task within the time budget. The 4 robot team actually completes the task most often, but has the highest average final entropy in the target estimate, indicating that it is the least consistent. For this environment and these robots, the 2 robot team seems to perform best on average: the time to completion is competitive in all scenarios and it has the best average final entropy in the 4 target case.

We also tested the system in a larger $4 \times 4\,\mathrm{m}$ environment. We kept all other system parameters the same, including the location of the targets and the initial positions of the robots. In this larger environment we saw a different trend, with the 4 robot team finishing the mission the fastest, as Fig. 9 shows. The larger environment (4 times the area of the original) requires more time for the robots to traverse, giving larger teams the advantage. This additional space also allows the robots to spread out and explore disjoint areas, decoupling the

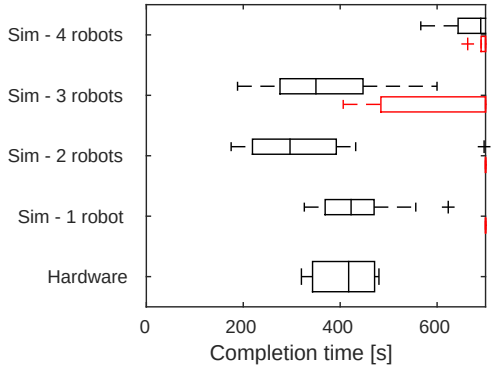
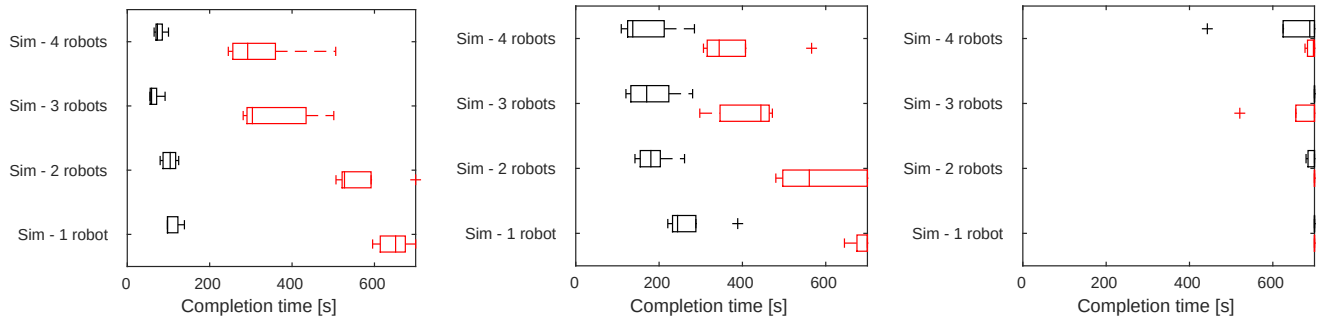


Fig. 8. Box plots of the time to completion for the simulated and hardware MAD experiments with 2 targets. Black lines are the in small environment where the hardware experiments were conducted. Red lines are the larger simulated environment.

TABLE II
AVERAGE PERCENTAGE OF TIME SPENT COMPUTING.

		Number of targets							
		Small environment				Large environment			
		0	1	2	4	0	1	2	4
# robots	1	0.8	1.5	5.4	15.9	0.5	1.5	5.6	12.7
	2	0.9	3.0	14.9	41.2	0.6	2.5	10.0	24.0
	3	2.0	8.3	35.4	69.6	1.0	4.6	15.9	46.7
	4	5.7	17.4	48.2	84.9	1.2	5.0	20.7	51.7

control computations and decreasing the number of times the collision avoidance algorithm is activated. We see the effects of this in Table II, with the computation time growing more slowly with the team size.



(a) 0 target simulations (b) 1 target simulations (c) 4 target simulations

Fig. 9. Box plots of the time to completion for simulated experiments using teams of different sizes in environments with a varying number of targets. The small environment is in black and the large environment is in red.

TABLE III
AVERAGE FINAL ENTROPY [BITS].

		Number of targets							
		Small environment				Large environment			
		0	1	2	4	0	1	2	4
# robots	1	0.08	0.09	0.08	2.78	0.27	0.55	3.94	7.72
	2	0.09	0.09	0.08	1.49	0.28	0.37	2.30	7.07
	3	0.07	0.06	0.07	3.49	0.09	0.08	1.15	8.20
	4	0.09	0.09	1.43	8.96	0.09	0.10	1.35	6.93

IV. CONCLUSIONS

Magnetic anomaly detection is an important task in military surveillance, humanitarian de-mining efforts, geological mapping, and more. In this paper we present a unified control and estimation framework to autonomously discover, count, and localize an unknown number of targets. We present a series of hardware and simulated experiments using a small team of quadrotor MAVs. The MAVs are equipped with magnetometers, which detect the presence of nearby magnetic sources, providing only a single bit of data. Despite this very coarse sensor, the robots are able to successfully determine the number of targets and to localize each target with a precision at the length scale of the targets.

This framework may also be applied to other scenarios with noisy, binary sensors, such as touch sensors or any sensor where the data is thresholded. This work also applies to robots other than MAVs, though robots with kinematic constraints may need to project the gradient onto the set of feasible control inputs. Future work will focus on planning over longer time horizons, planning for robots with multiple sensors attached to each platform, and in situations where robots must actively sense and avoid hazards.

REFERENCES

[1] T. R. Clem, "Sensor technologies for hunting buried sea mines," in *MTS/IEEE OCEANS*, vol. 1. IEEE, 2002, pp. 452–460.
[2] P. Gunn and M. Dentith, "Magnetic responses associated with mineral deposits," *AGSO Journal of Australian Geology and Geophysics*, vol. 17, pp. 145–158, 1997.
[3] M. J. Caruso, T. Bratland, C. H. Smith, and R. Schneider, "A new perspective on magnetic field sensing," *Sensors*, vol. 15, pp. 34–47, 1998.
[4] A. Sheinker, L. Frumkis, B. Ginzburg, N. Salomonski, and B.-Z. Kaplan, "Magnetic anomaly detection using a three-axis magnetometer," *IEEE Trans. Magn.*, vol. 45, no. 1, pp. 160–167, 2009.
[5] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House Boston, 2007, vol. 685.
[6] P. Dames, P. Tokekar, and V. Kumar, "Detecting, Localizing, and Tracking an Unknown Number of Moving Targets Using a Team of Mobile Robots," in *Intl. Sym. Robot. Research*, Sep 2015.
[7] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "Random Finite Sets for Robot Mapping and SLAM," *Springer Tracts in Advanced Robotics*, 2011.
[8] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas, "Semantic Localization Via the Matrix Permanent," in *Robotics: Science and Systems*, 2014.
[9] M. Schwager, P. Dames, D. Rus, and V. Kumar, "A Multi-Robot Control Policy for Information Gathering in the Presence of Unknown Hazards," in *Intl. Sym. Robot. Research*, August 2011.
[10] P. Dames, M. Schwager, V. Kumar, and D. Rus, "A Decentralized Control Policy for Adaptive Information Gathering in Hazardous Environments," in *IEEE Intl. Conf. on Decision and Control*, Dec. 2012, pp. 2807–2813.
[11] B. Ristic and B.-N. Vo, "Sensor Control for Multi-Object State-Space Estimation Using Random Finite Sets," *Automatica*, vol. 46, no. 11, pp. 1812–1818, Nov. 2010.
[12] B. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed Robotic Sensor Networks: An Information-Theoretic Approach," *Intl. J. Robot. Research*, vol. 31, no. 10, pp. 1134–1154, Aug. 2012.
[13] R. A. Cortez, H. G. Tanner, R. Lumia, and C. T. Abdallah, "Information Surfing for Radiation Map Building," *International Journal of Robotics and Automation*, vol. 26, no. 1, p. 4, 2011.
[14] B. Grocholsky, "Information-Theoretic Control of Multiple Sensor Platforms," Ph.D. dissertation, University of Sydney, 2002.
[15] N. Atanasov, J. Le Ny, N. Michael, and G. J. Pappas, "Stochastic source seeking in complex environments," in *IEEE Intl. Conf. on Rob. and Autom.* IEEE, 2012, pp. 3013–3018.
[16] B. Charrow, V. Kumar, and N. Michael, "Approximate Representations for Multi-Robot Control Policies that Maximize Mutual Information," *Robotics: Science and Systems*, 2013.
[17] P. Dames, D. Thakur, M. Schwager, and V. Kumar, "Playing Fetch With Your Robot: The Ability of Robots to Locate and Interact with Objects," *IEEE Robot. Autom. Mag.*, vol. 21, no. 2, pp. 46–52, 2014.
[18] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
[19] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Intl. J. Robot. Research*, vol. 33, no. 9, Aug. 2014.
[20] G. Hollinger, S. Yerramalli, S. Singh, U. Mitra, and G. S. Sukhatme, "Distributed data fusion for multirobot search," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 55–66, 2015.
[21] Z. W. Lim, D. Hsu, and W. S. Lee, "Adaptive informative path planning in metric spaces," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 283–300.
[22] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Systems Technical Journal*, vol. 27, pp. 379–423, 1948.
[23] M. Angermann, M. Frassl, M. Doniec, B. J. Julian, and P. Robertson, "Characterization of the indoor magnetic field for applications in localization and mapping," in *IEEE Intl. Conf. on Indoor Positioning and Indoor Navigation*. IEEE, 2012, pp. 1–9.