Multiscale Adaptive Search

Alice Hubenko, Vladimir A. Fonoberov, George Mathew, and Igor Mezić, *Member, IEEE*

***Abstract*—We present a continuous-space multiscale adaptive search (MAS) algorithm for single or multiple searchers that finds a stationary target in the presence of uncertainty in sensor diameter. The considered uncertainty simulates the influence of the changing environment and terrain as well as adversarial actions that can occur in practical applications. When available, information about the foliage areas and *a priori* distribution of the target position is included in the MAS algorithm. By adapting to various uncertainties, MAS algorithm reduces the median search time to find the target with a probability of detection of at least P_D and a probability of false alarm of at most P_{FA}. We prove that MAS algorithm discovers the target with the desired performance bounds P_D and P_{FA}. The unique features of the MAS algorithm are realistic second-order dynamics of the mobile sensors that guarantees uniform coverage of the surveyed area and a two-step Neyman–Pearson-based decision-making process. Computer simulations show that MAS algorithm performs significantly better than lawnmower-type search and billiard-type random search. Our tests suggest that the median search time in the MAS algorithm may be inversely proportional to the number of participating searchers. As opposed to lawnmower search, the median search time in the MAS algorithm depends only logarithmically on the magnitude of uncertainty.**

***Index Terms*—Area coverage, detection algorithms, false alarm, probability, search theory, target, uncertainty, unmanned aerial vehicles.**

I. Introduction

STUDY of search problems using formalized mathematical models started more than 60 years ago. For a survey, see [1]. During World War II, mathematical theory was applied for the first time to locate German submarine threats in the Atlantic. Since then, the mathematics behind the practical search problems developed into search theory. In the "classical" discrete-space setting (see e.g., [2]–[8]), a target is located somewhere in a region that is partitioned into a finite number of cells. The probability distribution for the target's position (i.e., the probability that the target is in any particular cell) and the detection function of a sensor (i.e., the probability of detection versus effort spent searching a cell, given that the target resides in that cell) are given. The goal is to maximize the probability of detection of the target, assuming that the amount of total effort available for the search is fixed. A major drawback of the above setting is its discrete nature and the assumption of perfectly functioning sensors. Moreover, theoretical solutions given to the discrete space search problem assume that it is possible to move between any two cells and, thus, result in trajectories that could be physically impossible to follow.

An alternative to the discrete search setting is the "natural" continuous search setting developed in [9] and [10]. The theory of continuous search [9], [10] is more suitable for practical applications, as it can be extended to include various uncertainties in the sensor, environment, etc. However, quoting from [10]: "It is hard to imagine a sensor track, which is randomly but uniformly distributed in the rectangle." In the following, we will show that such sensor tracks can be constructed for any number of searches.

Recently, several application-oriented search algorithms have been developed [11]–[17]. In [11] and [18], a methodology is developed to deploy a mobile sensor network for the purpose of detecting and capturing a mobile target that is moving in a straight line on the plane. The dynamics of the pursuers is limited to a combination of circles and lines (non-holonomic unicycle model). The authors use track before detect approach described in [19] where the stationary proximity sensors detect the track of the target before pursuing it. In [12], an undersea collaborative search framework is described where the sensors move in the search area divided into cells. The sensors take measurements within a cell multiple times to achieve the desired level of assurance that the target is not within the cell. In [20], cooperative track detection performance is optimized by deploying the sensors based on their future displacement, which can be estimated from environmental forecasts and sensor dynamic models. The target is assumed to move with constant speed on a straight line. In [13], [14], authors present a receding-horizon cooperative search algorithm that jointly optimizes routes and sensor orientations for a team of autonomous agents searching for a mobile target. The algorithm in [13] reduces the continuous search problem to an optimization on a finite graph. In [15], a framework for cooperative search using unmanned aerial vehicle (UAV) swarms is described. The algorithm in [15] sweeps the area with UAV's flying side-by-side in straight lines. In [16], [17], an entropy-based algorithm for search and action mission is presented. Continuous agent search paths, resulting from the algorithm in [17], overlap, reducing search efficiency. None of the state-of-the-art algorithms described in this paragraph take into account uncertainty in terrain (e.g., areas covered with ice or snow), environment (e.g., wind or fog), or sensor malfunction (e.g., wrong altitude or complete loss of some agents). Different sources of uncertainty

Manuscript received February 6, 2010; revised September 5, 2010 and November 28, 2010; accepted December 20, 2010. Date of publication February 7, 2011; date of current version July 20, 2011. This work was supported in part by DARPA DSO under AFOSR Contract FA9550-07-C-0024 and ONR Grant N00014-10-1-0544. This paper was recommended by Associate Editor S. Ferrari.

A. Hubenko and I. Mezić are with the Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: hubenko@engineering.ucsb.edu; mezic@engineering.ucsb.edu).

V. A. Fonoberov is with AIMdyn, Inc., Santa Barbara, CA 93101 USA (e-mail: vfonoberov@aimdyn.com).

G. Mathew was with the Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106 USA. He is now with United Technologies Research Center (UTRC), Berkeley, CA 94705 USA (e-mail: mathewga@utrc.utc.com).

Digital Object Identifier 10.1109/TSMCB.2011.2106207

alter the effective diameter of the sensor. This would leave parts of the search area completely uncovered and further reduce the performance of the search algorithms.

We consider the search problem where a stationary target is placed in an area S that contains foliage F that the sensors cannot penetrate. The *a priori* distribution of the location of the target (prior function) is assumed to be given. If it is not given, we assume it to be the uniform distribution. The mobile sensors (searchers) move through the area S continuously and use a sensor with a circular range to scan the area. Our goal is to minimize the median search time in the presence of uncertainty in sensor diameter while keeping the probability of detection of our algorithm above a threshold P_D and the probability of false alarm of the algorithm below a threshold P_{FA}. Uncertainties in terrain, environment as well as adversarial actions can be considered as special cases of uncertainty in sensor diameter.

The multiscale adaptive search (MAS) algorithm developed here is a fusion of uniform-coverage dynamics of the sensors and a two-step decision-making algorithm. When in the explore mode, the sensors move under the spectral multiscale coverage (SMC) dynamics developed in [21] and described in the Appendix. SMC dynamics ensures that the trajectories of the sensors are uniformly spaced throughout the search area and thus making it difficult for a target to evade detection by the sensors. Realistic second-order dynamics of the sensors is employed. The Neyman–Pearson lemma [22], that is central in binary hypothesis testing theory to design a decision-making rule, allows the sensors to quickly locate target suspects. Our two-step Neyman–Pearson decision making puts some of the sensors into rechecking mode to take additional measurements at suspect target positions. This strategy ensures that the probability of false alarm is within the required threshold. It is the combination of the SMC dynamics and the two-step Neyman–Pearson-based decision making that makes our MAS algorithm perform very well under various uncertainties. The MAS algorithm is scalable to very large number of sensors n because the computational complexity is at most $n\log(n)$ (which is explained in Section IV). In the case of a non-uniform prior, we demonstrate that the median search time is minimized when the sensor's trajectories are designed according to the logarithm of the prior function.

We tested the MAS algorithm with 50 sensors for different *a priori* target distributions, each time making 5000 independent simulations. Our computer simulations show that besides demonstrating superior robustness in the presence of uncertainty, the MAS algorithm vastly outperforms random search algorithms, such as the billiard-type search described in Section IV, see also [23], where sensors start out in random directions, move in straight lines, and reflect when they reach the border. Another important advantage of the MAS algorithm is its effective use of assets: The search time is inversely proportional to the number of sensors. We also find that the search time of the MAS algorithms is proportional to the logarithm of the magnitude of uncertainty, whereas the best design of lawnmower-type algorithms described in Section VI, see also [24], result in linear dependence of the search time on the magnitude of uncertainty. In the following, we present a detailed description of the theory behind MAS algorithm and demonstrate its performance for various applications (Table I).

TABLE I
SUMMARY OF PARAMETERS AND VARIABLES

n	number of sensors (searchers)
n_0	number of measurements to define target suspects
γ_0	number of detections to define target suspects
n_1	number of measurements for the re-check steps
γ_1	number of detections for the re-check steps
s_d	prob. of detection for a single measurement
s_{fa}	prob. of false alarm for a single measurement
p_{md}	prob. of missed detection per n_0 measurements
p_{fa}	prob. of false alarm per n_0 measurements
P_{MD}	$= 1 - P_D$ prob. of missed detection of the algorithm
P_{FA}	prob. of false alarm of the algorithm
f	frequency of the sensor measurements
α	prob. for target to be in foliage $(0 \leq \alpha < 1)$
δ	upper bound of the radius of the sensor
S	the search area
F	foliage
T_{stop}	search stopping time

II. MAS: THE DECISION-MAKING ALGORITHM

Assume that we have n sensors and an area S where a single target is located. S may contain foliage F which the sensors cannot penetrate. If the target is in the foliage, it is undetectable for the sensors. The goal is to design a search algorithm that will detect the target with probability of false alarm at most P_{FA} and probability of missed detection at most P_{MD}. Next, we describe our decision-making algorithm with uniform coverage dynamics (MAS search); it can be used with any coverage strategy. Depending on the coverage algorithm, the stopping time is adjusted to achieve the desired precision. To create a measurement history map and keep track of the sensor measurements made in each cell area, S is divided into small enough cells. There are two main modes for each sensor, explore and recheck.

1) For given P_{FA} and P_{MD}, s_{fa}, and s_d, we compute the constants n_0 and γ_0. Later in this section, we will discuss in detail how to find the above constants using the Neyman–Pearson lemma.
2) For given sensor frequency f and estimate of search stopping time T_{stop}, find p_{fa} and p_{md}, the expected probabilities of false alarm and missed detection per n_0 measurements, respectively.
3) For given p_{fa} and p_{md}, s_{fa}, and s_d, we compute the constants n_1 and γ_1. Detailed discussion is given later in this section.
4) Initialize measurement history map and list of target suspects to zero.
5) Deploy sensors. In our examples, sensors are deployed from a small grid on lower left corner of S. After deployment, all sensors start out in explore mode.
6) In explore mode, the sensors cover the search area using uniform coverage dynamics (discussed in Section V) and update the measurement history map.
7) If the number of detections at a location exceeds γ_0 and the number of measurements at the location does not exceed n_0, the location is added to the list of target suspects.

8) Starting from the most likely targets (locations that have the highest ratio of positive measurements), each target suspect is assigned to an available neighboring sensor in explore mode. The sensor that has been assigned a target suspect changes his mode to rechecking and moves to the location of the target on a straight line with maximum speed.
9) In recheck mode, the sensor performs n_0 measurements flying slowly above a target suspect position then switches to explore mode.
10) If the ratio of detections to the number of measurements exceeds γ_1/n_1, keep the location in the list of target suspects, otherwise remove it from the list.
11) When the number of measurements at a location becomes n_1, check if the number of detections at the location exceeds γ_1. If yes, declare that the target is found and stop the search.
12) If the algorithm reaches stopping time T_{stop} without declaring a detection, the algorithm declares that the target is in the foliage.
13) Repeat steps 4) to 12) to get statistics of search times.
14) Repeat steps 2) to 13) to adjust the stopping time T_{stop}, if needed.

The simplified scheme of the MAS algorithm is illustrated in Fig. 1.

The initial estimate for the stopping time for the MAS search algorithm can be calculated using the following equation, which was obtained empirically:

$$T_{\text{stop}} = \beta \frac{|S|}{\delta^2} \log\left(\frac{1}{\max(P_{FA}+P_{MD},\alpha)}\right) \tag{1}$$

where constant β depends on the coverage algorithm of the sensor as it moves in S. Next, we explain the heuristics behind formula (1).

Our simulations show, see also Fig. 2, that the tail of the distribution of detection times (for all settings) can be fitted by the following exponential function shifted in time to account for deployment

$$f(t) = \frac{1}{t_m} e^{-\frac{t}{t_m}} \tag{2}$$

where t_m is a constant proportional to $|S|/\delta^2$. Note, that to obtain the formula for the stopping time, small search times are not relevant. Assume that $f(t)$ is our probability distribution. Then, the corresponding cumulative distribution function $F(t) = 1 - e^{-(t/t_m)}$. We can find the stopping time $t = T_{\text{stop}}/\beta$ by solving

$$F(t) = \min\left(1-(P_{FA}+P_{MD}), 1-\alpha\right)$$

which is equivalent to

$$1 - e^{-\frac{t}{t_m}} = \min\left(1-(P_{FA}+P_{MD}), 1-\alpha\right).$$

Thus,

$$t = t_m \log\left(\frac{1}{\max(P_{FA}+P_{MD},\alpha)}\right).$$

Empirically, we obtained that for MAS search $\beta = 2.3$.

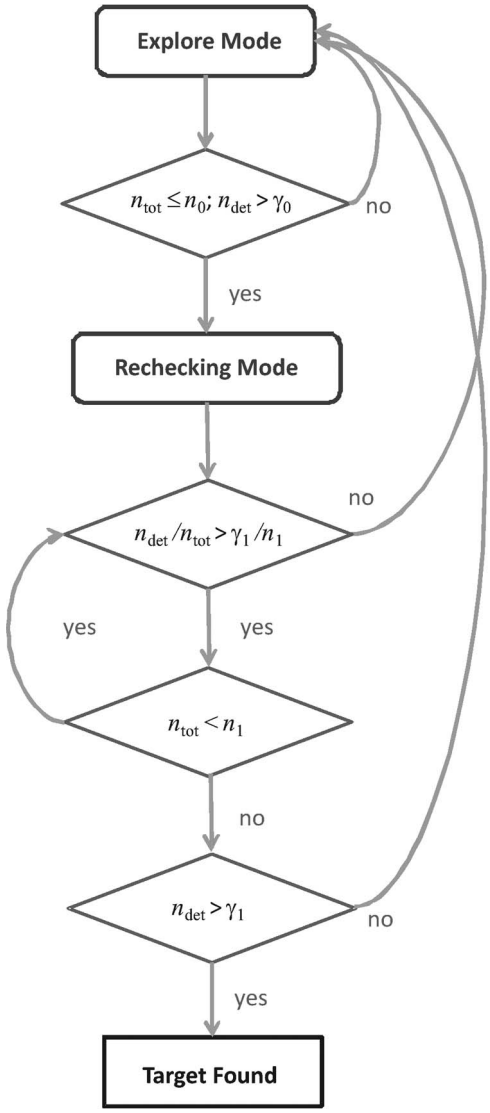

Fig. 1. Simplified scheme of the MAS algorithm. n_{tot} denotes the total number of measurements taken at a given location and n_{det} denotes the number of positive readings (detections) obtained at this location.

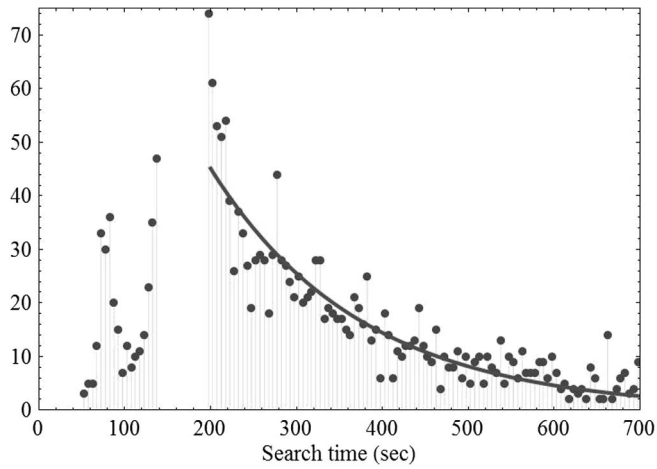

Fig. 2. Fitting the tail of the distribution of stopping times (example from Section IV). The solid curve is a fit given by (2).

In general, the stopping time of the algorithm can be adjusted in various ways. One way is to obtain statistics of stopping times of the algorithm, then compute the corresponding values of P_{FA}^T and P_{MD}^T based on statistics for each stopping time T. Now, the stopping time for the desired P_{FA} and P_{MD} can be extrapolated.

Next, we describe how to find the constants n_0, γ_0, n_1, and γ_1 used in our algorithm.

Recall that P_{MD} is the probability of declaring that the target is in foliage, given that the target is in S and P_{FA} is the probability of declaring that the target is somewhere in S, given that the target is not in that location. In the simulation setting, it translates to the following, as seen in [25]. Let N_R denote the number of realizations of the whole search scenario, N_{FA}—the number of times the algorithm declared finding a target given that the target was not there, N_D—the number of times the target was detectable (in S), and N_{MD}—the number of times the target was detectable and the algorithm declared that it was in the foliage. Then, we have

$$P_{FA} = \lim_{N_R\to\infty} \frac{N_{FA}}{N_R}$$
$$P_{MD} = \lim_{N_R\to\infty} \frac{N_{MD}}{N_D}.$$

In course of the algorithm, the sensor moves in S, taking measurements with frequency f. For an easier description of our decision-making procedure, let us group several measurements in one location and consider it a step. We chose this name as an analogy to a discrete search scenario where in each step, the sensor is allowed to make several independent measurements (at that location). Assume that each step includes n_0 independent measurements and detection of target is declared if at least $\gamma_0 + 1$ of the measurements are 1 s. The Neyman–Pearson criterion (see [22]) allows us to find n_0 and γ_0 that maximize the probability of detection while the probability of false alarm stays under some prescribed bound (P_{FA}). The Neyman–Pearson lemma (see [22]) implies that the minimal n_0 and γ_0 are the solutions to the following optimization problem:

$$\begin{aligned} P_{FA} &= P[k > \gamma_0] + \rho P[k = \gamma_0] \\ &= \sum_{k=\gamma_0+1}^{n_0} \binom{n_0}{k} s_{fa}^k (1 - s_{fa})^{n_0-k} \\ &\quad + \rho \binom{n_0}{\gamma_0} s_{fa}^{\gamma_0} (1 - s_{fa})^{n_0-\gamma_0} \qquad (3) \\ 1 - P_{MD} &= P[k > \gamma_0] + \rho P[k = \gamma_0] \\ &= \sum_{k=\gamma_0+1}^{n_0} \binom{n_0}{k} s_d^k (1 - s_d)^{n_0-k} \\ &\quad + \rho \binom{n_0}{\gamma_0} s_d^{\gamma_0} (1 - s_d)^{n_0-\gamma_0}. \qquad (4) \end{aligned}$$

We first find minimal γ_0 satisfying (3) when $\rho = 0$. Because at this point n_0 is unknown, $\gamma_0 = \gamma_0(n_0)$ is a function of n_0. Next, from the (3), we find $\rho = \rho(n_0)$. Finally, we substitute $\gamma_0(n_0)$ and $\rho(n_0)$ into (4) and find the minimal n_0 for which the equation still holds. Taking n_0 measurements at each location guarantees that probability of missed detection of the algorithm will be less than or equal to P_{MD}. It does not however guarantee that the probability of false alarm of the algorithm is less than or equal to P_{FA}. Taking n_0 measurements will be a preliminary criteria in our decision-making algorithm: If at least $\gamma_0 + 1$ readings are 1 s, the sensor will assume that there is a target suspect at that location. To achieve probability of false alarm less than or equal to P_{FA}, the sensors will take additional measurements.

Denoting the total number of steps taken by N, we can express the upper bound p_{fa} for probability of false alarm for each step as follows:

$$(1 - p_{fa})^N = 1 - P_{FA} \qquad (5)$$

where N can be estimated as $fT_{\text{stop}}/1.14n_0$. From (5), we get

$$p_{fa} = 1 - (1 - P_{FA})^{\frac{1.14n_0}{fT_{\text{stop}}}}. \qquad (6)$$

p_{fa} provides an upper bound for the probability of false alarm for each step needed for the algorithm to achieve probability of false alarm at most P_{FA}. The probability of missed detection for one step is the probability of declaring that the target is in foliage when the target is in S. Denoting by p_{md}, the upper bound for probability of missed detection for each step, we get

$$p_{md} = 1 - (1 - P_{MD})^{\frac{1.14n_0}{fT_{\text{stop}}}}. \qquad (7)$$

p_{md} provides an upper bound for the probability of missed detection for each step. Using the Neyman–Pearson criterion again, we obtain the constants n_1 and γ_1, that will be used by the algorithm in making the final decision. n_1 will be the upper bound on the number of measurements that the sensor may take at one step. With given p_{fa} and p_{md}, we find optimal n_1 and γ_1 as follows. From Neyman–Pearson lemma (see [22]), we have

$$\begin{aligned} p_{fa} &= P[k > \gamma_1] + \rho P[k = \gamma_1] \\ &= \sum_{k=\gamma_1+1}^{n_1} \binom{n_1}{k} s_{fa}^k (1 - s_{fa})^{n_1-k} \\ &\quad + \rho \binom{n_1}{\gamma_1} s_{fa}^{\gamma_1} (1 - s_{fa})^{n_1-\gamma_1} \qquad (8) \\ 1 - p_{md} &= P[k > \gamma_1] + \rho P[k = \gamma_1] \\ &= \sum_{k=\gamma_1+1}^{n_1} \binom{n_1}{k} s_d^k (1 - s_d)^{n_1-k} \\ &\quad + \rho \binom{n_1}{\gamma_1} s_d^{\gamma_1} (1 - s_d)^{n_1-\gamma_1}. \qquad (9) \end{aligned}$$

To find n_1, we find minimal $\gamma_1 = \gamma_1(n_1)$ satisfying (8) when $\rho = 0$. Next, from the (9), we find $\rho = \rho(n_1)$. Finally, we substitute γ_1 and $\rho(n_1)$ into (9) and find the minimal n_1 for which the inequality still holds.

III. Increasing Efficiency by Modifying the Prior

In [26], it was shown that for random search, the probability of detecting the target at (x, y) given the density function

of search effort $\phi(x,y)$ can be described by the following exponential law of random search

$$D\left(x,y,\phi(x,y)\right) = 1 - \exp\left[w(x,y)\phi(x,y)\right].$$

Here, $w(x,y)$ can be regarded as a local measure of detectability at (x,y). Note, that if the sensor sweeps the area with sensor of a fixed range $w(x,y) = W$ is assumed to be constant. Denote $p(x,y)$ the *a priori* probability distribution of the position of the target. To optimize the search effort in [26], it was proved that for an optimal $\phi(x,y)$, there is a constant $\lambda > 0$ such that

- if $\phi(x,y) > 0$, $p(x,y)w(x,y)\exp[w(x,y)\phi(x,y)] = \lambda$,
- if $\phi(x,y) = 0$, $p(x,y)w(x,y) \leq \lambda$.

Denote $A(\Phi) = \{(x,y) : \phi(x,y) > 0\}$. In the above terms for $(x,y) \in A(\Phi)$, the optimal $\phi(x,y)$ can be expressed as

$$\phi(x,y) = \frac{1}{w(x,y)} \log \frac{p(x,y)w(x,y)}{\lambda}. \tag{10}$$

Since, Φ, the total search effort is the integral of $\phi(x,y)$ over $A(\Phi)$, from (10), it follows that to optimize $\phi(x,y)$, one has to find the value of λ that will result in

$$\iint\limits_{A(\Phi)} \phi(x,y)dxdy = \Phi.$$

Geometrically, this can be interpreted as cutting a horizontal stripe from the graph of $\phi(x,y)$ up till $y = \log\lambda$. We show that without the constraints of "random search", a similar approach can be used to increase the search efficiency.

Denote by $p(t)$ the probability of the target being detected by time t. For a small interval of time δt

$$p(\delta t) = 1 - (1 - s_d)^{\delta t f} \tag{11}$$

where s_d is the probability of detecting the target by a single measurement and f is the frequency with which the sensor takes the measurements. Assuming that the sensor radius does not change with time, we can write $p(\delta t) = w(\delta t)\delta t$ with some continuous function $w(t)$.

$$\begin{aligned} w(0) &= \lim_{\delta t \to 0} \frac{1 - (1 - s_d)^{\delta t f}}{\delta t} \\ \left(1 - (1 - s_d)^{ft}\right)'(0) &= -f\log(1 - s_d)(1 - s_d)^{ft}(0) \\ &= -f\log(1 - s_d) \end{aligned} \tag{12}$$

so

$$w(0) = -f\log(1 - s_d). \tag{13}$$

Note, that exponential law is true for the non-detection probability. Denote by $ND(t)$ the probability of the event that the target have not been detected by time t. Then,

$$\begin{aligned} ND'(t) &= \lim_{\delta t \to 0} \frac{ND(t + \delta t) - ND(t)}{\delta t} \\ &= \lim_{\delta t \to 0} \frac{(1 - p(\delta t))\, ND(t) - ND(t)}{\delta t} \\ &= -w(0)ND(t) \\ &= f\log(1 - s_d)ND(t). \end{aligned} \tag{14}$$

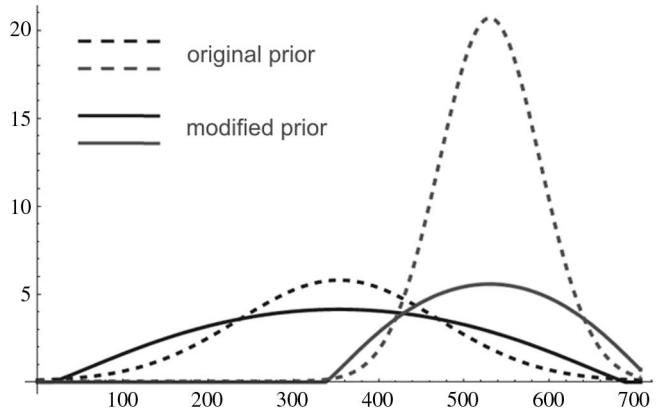

Fig. 3. Reduction of high peaks in $\mathcal{P}$ (dashed curve), by replacing it with $\log(\mathcal{P}/\lambda) > 0$ (solid curve).

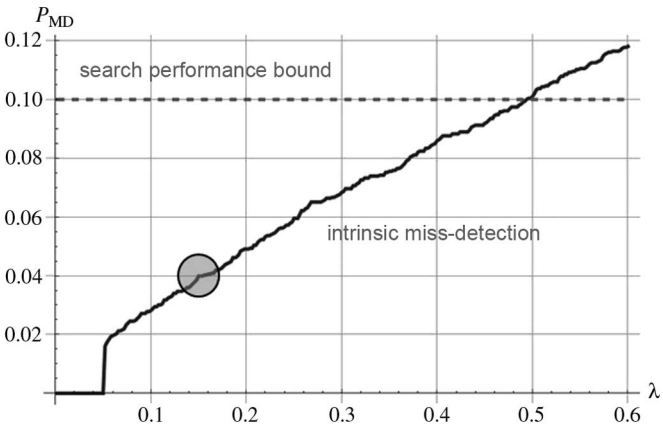

Fig. 4. Automatic procedure to select λ.

To prevent the sensor from covering unevenly the high probability areas, we modify the prior using logarithm and use a cutoff probability to prevent the sensors from going into areas where the probability of the target being there is close to zero. Fig. 3 illustrates how the high peaks are reduced by taking logarithm of the prior, or for short, log prior. In Fig. 4, the log prior distribution is analyzed for the non-uniform distribution from Fig. 5. The black circles around the peaks of the probability distribution in Fig. 6 show the place from where the log prior distribution is set to zero. Given the original prior $p(x,y)$, we automatically construct a log prior as follows. Every value $p(x,y)$ is replaced by

$$p(x,y) \to \max\left[\log\left(p(x,y)/\lambda\right), 0\right] \tag{15}$$

where λ is selected as shown below.

For a fixed value of λ, we define intrinsic miss detection as integral of $p(x,y)$ over the area where the log prior equals zero $\{(x,y) : \max[\log(p(x,y)/\lambda), 0] = 0\}$. Fig. 4 shows the intrinsic miss detection as a function of λ. Now, by setting the value of intrinsic miss detection sufficiently low, we can obtain λ. In Fig. 4, the grey circle shows the value of λ for which the intrinsic miss detection is 0.04, i.e., 4% of detectable targets will never be found (as they will be outside of two circular areas in Fig. 6). When the modified prior is used to compute MAS

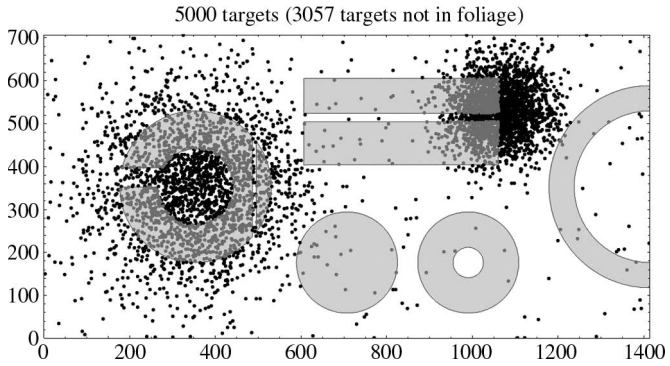

Fig. 5. Search area S with 5000 targets generated according to a non-uniform prior. A fraction of the targets is under foliage, which is shown in grey.

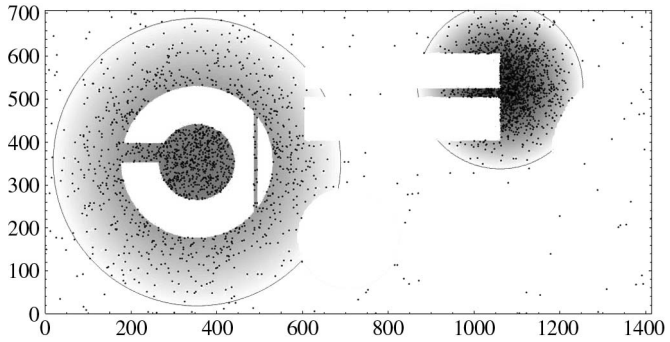

Fig. 6. Density of $\log(\mathcal{P}/\lambda) > 0$. Dots represent detectable targets.

trajectories, the sensors are confined to the areas where the target is likely to be located. The sensors are better distributed and they do not waste time going through the prior maxima over and over again.

IV. Performance of the MAS Search Algorithm

In Fig. 7, we present performance plots of our decision-making algorithm described in Section II. Fixing constants n_0, s_d, and s_{fa}, we can compute all corresponding pairs P_{FA}, P_{MD} using (3) and (4). For fixed s_d and s_{fa}, each color represents a constant n_0, shown on the picture. The pairs (s_d, s_{fa}) that we use are characteristics of realistic sensors computed in [27]. Fig. 7 illustrates the tradeoff between precision of the algorithm and the number of measurements it has to take. The smaller are the probabilities of P_{FA} and P_{MD}, the bigger is the n_0. Fig. 7 also shows that our decision-making algorithm is highly efficient, for reasonable choice of P_{FA} and $P_{MD}n_0$ stays fairly small. For example, the high-precision $P_{FA} = 0.1$ and $P_{MD} = 0.1$ can be achieved with n_0 between 3 and 5 when $s_d = 0.8$ and $s_{fa} = 0.2$.

Next, we show that the computational complexity of the MAS search algorithm is at most $n\log(n)$. It takes $O(n)$ operations to compute trajectories for the n sensors as described in Section V, see [21]. The algorithm has to update trajectories every time after a sensor switches into recheck mode which still results in $O(n)$ computational complexity, since there is an upper bound on the search time T_{stop}. When a sensor has to be assigned to recheck a target suspect, the algorithm performs a sorting of sensors according to their distance to the target suspect to find the closest available sensor. Since the complexity of a simple sorting algorithm is $O(n\log(n))$, see [9], the computational complexity of the MAS algorithm is at most $O(n\log(n))$.

We tested the MAS search algorithm for 50 sensors on a rectangular area S shown in Fig. 5. The sensors are deployed in vertices of a 10×5 small grid in lower left corner of S. Foliage is shown in light grey and the prior distribution is shown by dots. The sensors move with realistic second- order dynamics (max velocity $= 10$ m/s, max acceleration $= 5$ m/s^2). Each sensor has an uncertain circular sensor range with diameter uniformly distributed between 5 and 10 m that changes every 50 s. Sensor frequency $f = 2$ Hz. The probabilities of detection and false alarm for a single sensor are $s_d = 0.8$ and $s_{fa} = 0.2$, respectively. The goal is to minimize search time while satisfying the requirements $P_{D,\text{group}} \geq 0.9$ and $P_{FA,\text{group}} \leq 0.1$. We tested the MAS algorithm with randomly generated targets drawn from the uniform prior, shown in Fig. 8, and with random targets drawn from a non-uniform prior, shown in Fig. 5. When the sensors move according to MAS coverage strategy, it is guaranteed that the distribution of points along the trajectories approaches the desired probability distribution of the prior. So, when the prior is non-uniform and the probability distribution is concentrated around several sharp peaks, the sensors using MAS coverage will spend much time in high probability areas close to the peaks and the areas with lower probability will get covered much slower. Moreover, if the peaks are sharp enough, the sensors will continue taking measurements even after enough measurements have been taken to decide that the target is not there.

Figs. 9–11 show typical coverage of the search area by 50 sensors in cases of uniform, non-uniform, and log non-uniform priors, respectively, for MAS search. Fig. 9 illustrates that the MAS algorithm provides a good coverage of the search area. As compared to the coverage of non-uniform prior in Fig. 10, coverage of log prior in Fig. 11 is more contained to high probability areas and covers them more efficiently.

We performed 5000 experiments for uniform, non-uniform, and log non-uniform priors, respectively, to obtain statistics of MAS search. In Fig. 12 are histograms of MAS search performed on S with non-uniform prior and with log non-uniform prior. The histograms for the cases of both non-uniform and log non-uniform priors have a high peak close to median detection time and a high single bin at the end of the distribution. The peak around median detection time can be explained by the fact that in the case of the non-uniform distributions, the targets are contained in relatively small areas with high probability and the sensors cover these areas by about the median detection time. Despite the fact that median detection times are similar for in cases of non-uniform prior (213 s) and log non-uniform prior (203 s), the median search time for the case of non-uniform prior is 15% bigger (459 s) than the the median search time for the case of log non-uniform prior (400 s). The median absolute deviation of MAS detection time applied with non-uniform prior is 40% bigger than when log non-uniform prior is used, while P_{MD} stays the same and P_{FA} is almost twice smaller for the log non-uniform prior. This means that the log

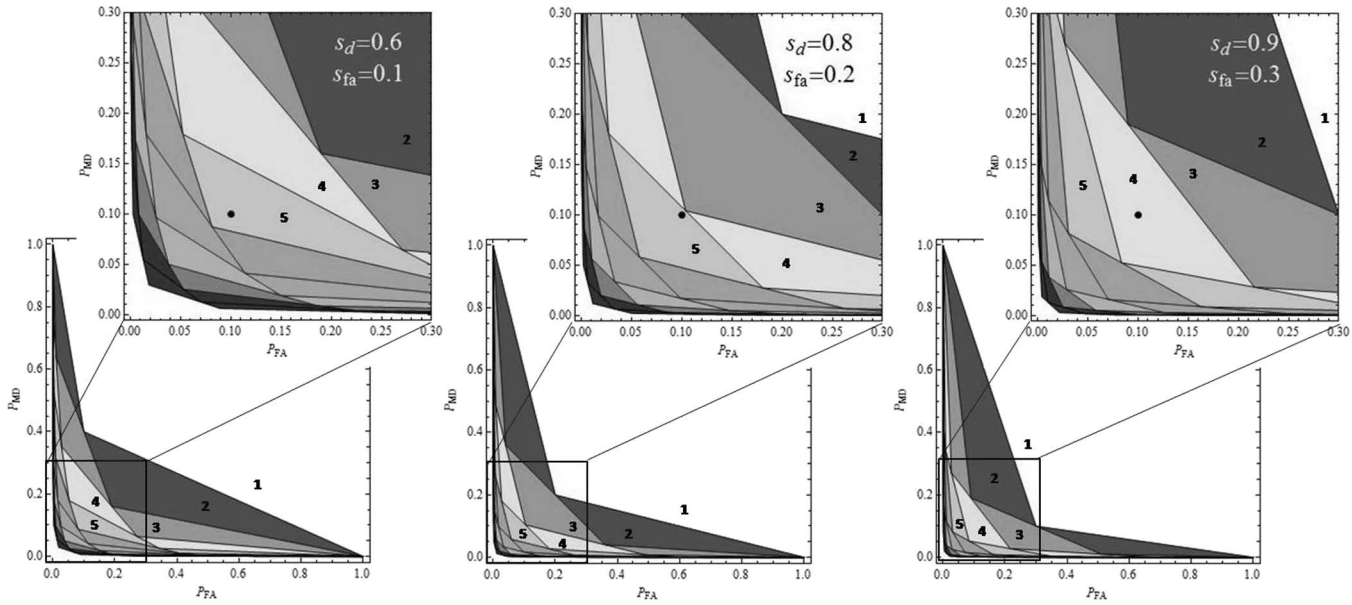

Fig. 7. Receiver operating characteristic curves in terms of P_{MD} and P_{FA}.

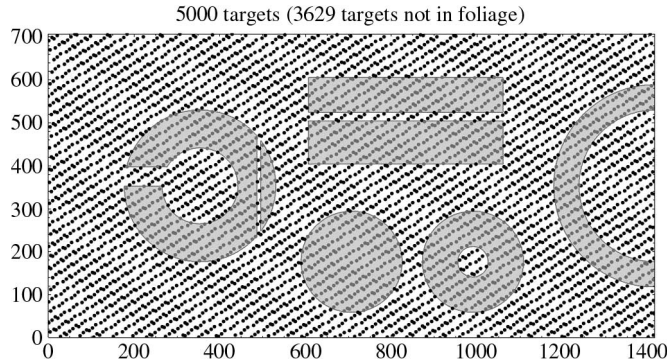

Fig. 8. Search area S with 5000 targets generated according to the uniform prior. A fraction of the targets is under foliage, which is shown in grey.

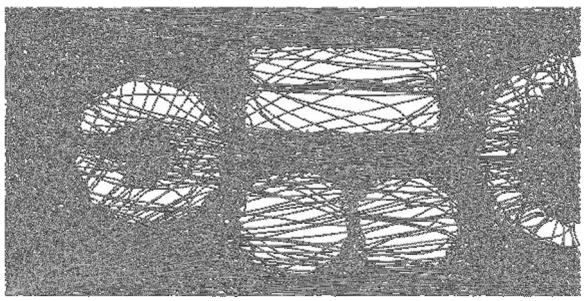

Fig. 9. Typical coverage of the uniform prior by 50 sensors.

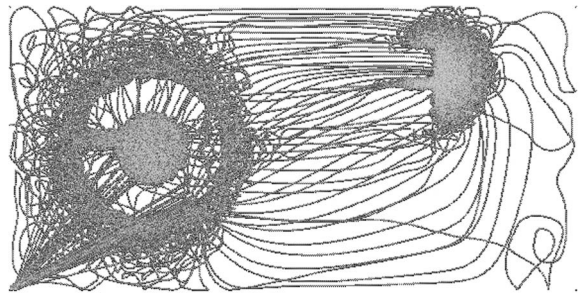

Fig. 10. Typical coverage of $\mathcal{P}$ by 50 sensors.

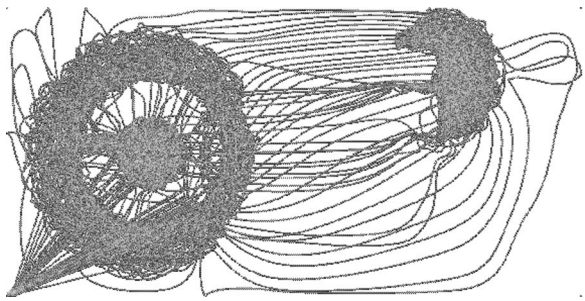

Fig. 11. Typical coverage of $\log(\mathcal{P}/\lambda) > 0$ by 50 sensors.

non-uniform prior results in more efficient coverage. When there are high peaks in the prior, we have seen that modifying the prior to log prior can significantly boost the performance of the MAS algorithm. The high peaks at the end of the distributions in Figs. 12 and 13 consist of the cases when the stopping time T_{stop} is reached, i.e., the algorithm declares that the target is in the foliage. The histogram of MAS search performed on S for uniform prior in Fig. 13 declines very slowly and does not contain high peaks except the single bin at the end of the distribution. This means that the coverage provided by the MAS algorithm is very homogeneous at all times which in turn means that the MAS search algorithm is very efficient.

Fig. 14 illustrates that $1 - P_{MD} = P_{D,\text{group}}$ and $P_{FA} = P_{FA,\text{group}}$ converges above and below the required limits, respectively, as the number of realizations of MAS search increases. So, our search algorithm, indeed, performs with the desired precision for uniform, non-uniform, and log non-uniform priors.

In Fig. 15, we can see the median search time converges as the number of realizations increases.

In Table II, we compare statistics for MAS search without uncertainty, MAS search with periodically changing uncertain

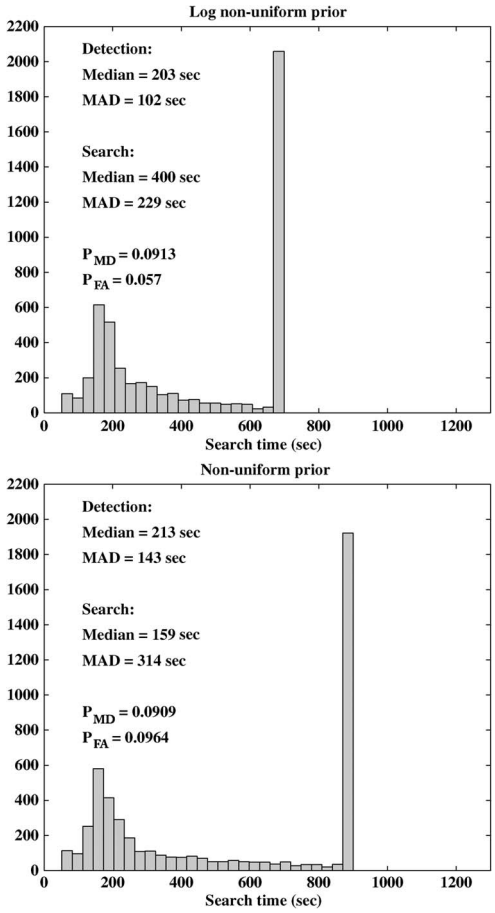

Fig. 12. Histograms of MAS search for non-uniform prior and log-non-uniform prior.

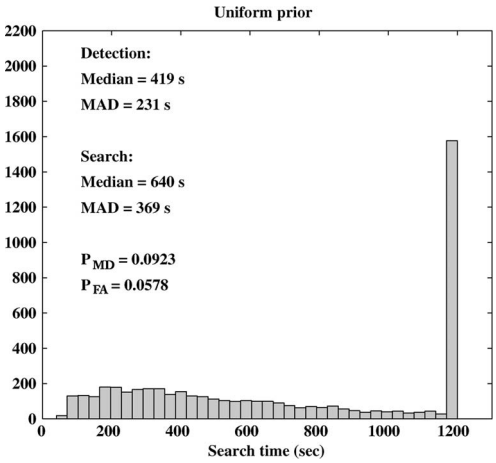

Fig. 13. Histogram of MAS search for uniform prior.

sensor radius, and billiard search when sensors start out in random directions and move in straight lines, reflecting when they reach the border. We ran computer simulations of MAS search conducting 5000 independent experiments for each scenario. Computer simulations show that median detection time, median search time, and median absolute deviation of MAS search in presence of uncertainty are very close to the corresponding MAS search data without uncertainty in sensor radius.

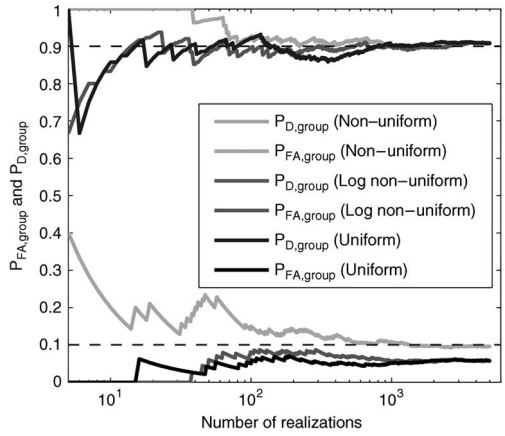

Fig. 14. $P_{D,\mathrm{group}}$ and $P_{FA,\mathrm{group}}$ obtained from our simulations statistics as the number of experiments increases.

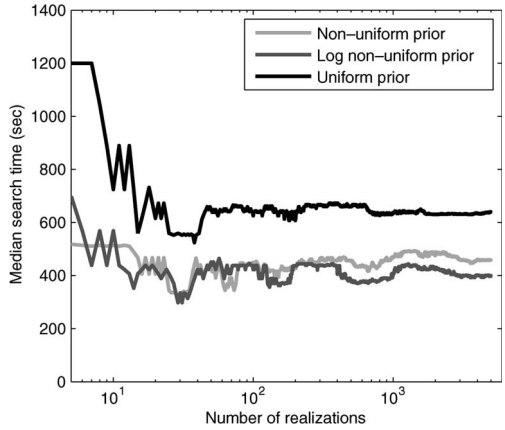

Fig. 15. Median search time of MAS algoithm.

The median absolute deviation of MAS search time is 1.5 times smaller than that of billiard search; median search time of MAS search is 1.6 times smaller than that of billiard search; median detection time of MAS search is 1.7 times smaller than that of billiard search.

V. Uniform Coverage Dynamics (SMC)

In MAS search, when the sensors are in search mode, they move according to the SMC algorithm (see [21] and [28]). This algorithm prescribes centralized feedback control laws for n sensors so that they achieve uniform coverage of the prescribed domain. The algorithm uses a metric that quantifies how far sensor trajectories are from being ergodic with respect to a given probability measure. This metric for ergodicity is a measure of how uniformly the points on the sensor trajectories cover a domain. The feedback controls applied on the moving sensors are essentially optimal controls that minimize the uniform coverage metric at the end of an infinitesimal time horizon. In the Appendix, we formally define the metric for ergodicity (or uniform coverage) and describe how the feedback laws for the motion of the sensors are computed. The algorithm in [21] allows to design optimal dynamics for sensors with the goal to cover any part of a given area S. For instance, if

TABLE II
COMPARISON OF MAS SEARCH UNDER DIFFERENT CONDITIONS

Algorithm	Median Detection Time	Median Search Time	Median Absolute Deviation
MAS (no uncertainty)	169 sec	303 sec	226 sec
MAS	203 sec	400 sec	229 sec
Billiard search	587 sec	1190 sec	534 sec

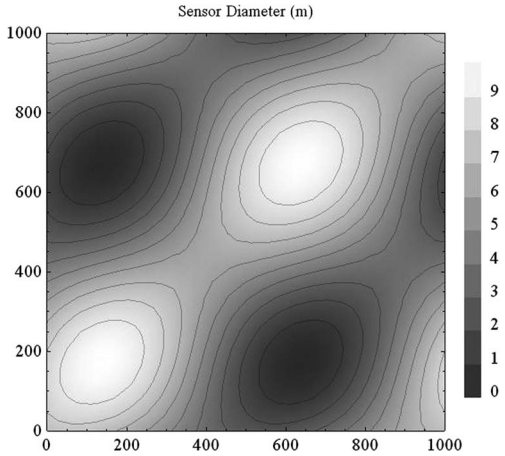

Fig. 16. Search area (1 km^2) with uncertain terrain. Sensor diameter as a function of position shown here is unknown to the sensors.

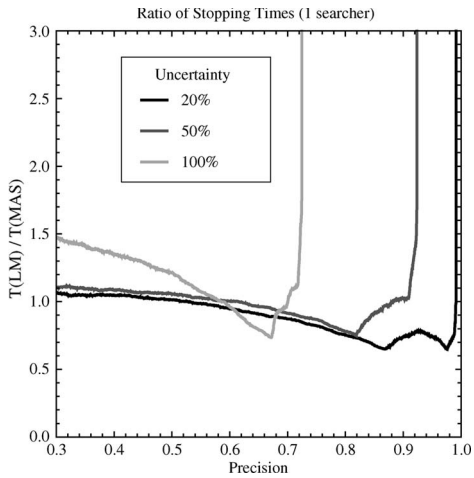

Fig. 17. Ratio of the search stopping times for lawnmower and MAS strategies as a function of the search precision (probability of the search success at stopping) for different magnitudes of terrain uncertainty.

the location of the target is described by a probability distribution $\mathcal{P}$, the part of S where $\mathcal{P} > 0$ has to be covered. We tested the uniform coverage dynamics used in MAS search, for 50 sensors and the area S shown in Fig. 5, where the foliage is shown in light grey. We tested the algorithm for the case when the prior probability distribution of the target $\mathcal{P}$ is uniform and for the case when the location of the target corresponds to a prior distribution shown in Fig. 5. As illustrated in Figs. 9 and 10, the uniform coverage motion design of the MAS algorithm guarantees superior coverage when the *a priori* distribution is both uniform and non-uniform. Note that the sensors move with realistic second-order dynamics and that the distribution of points on the trajectories approaches the *a priori* distribution. As seen in Fig. 10, the motion of the sensors depends on the *a priori* distribution: The high-probability regions are always covered better. By covering S according to a probability distribution, the uniform coverage dynamics saves time by not going to regions where the probability distribution of the target is zero.

VI. SEARCH WITH UNCERTAIN TERRAIN

Consider the search problem where the prior distribution of the target is uniform and the sensor range depends on the terrain but is unknown to the sensor, i.e., the terrain is uncertain. There is no foliage and all other parameters are the same as in Section IV. Fig. 16 is an example of an uncertain terrain in a 1000 $\times$ 1000 m square region that we will use in this section. Here, the sensor diameter varies between $D_{\min}$ and $D_{\max} = 10$ m, and the uncertainty in terrain is defined as $\epsilon = 1 - (D_{\min}/D_{\max})$. In Fig. 16, $D_{\min} = 0$ and the sensor diameter varies, depending on the location, from zero (shown in the darkest shade) to 10 m (shown in the lightest shade), so the terrain uncertainty is 100%. When terrain is uncertain, the best strategy for the sensor is to uniformly cover the entire search area. We will compare two coverage strategies fused with our decision-making algorithm: The lawnmower algorithm and the MAS algorithm. Here, we will use the version of the lawnmower coverage, in which the sensor is moving in the area in parallel lines and the locations where the sensor takes measurements form a triangular lattice. The distance between two neighboring lattice points is $r_{LM}\sqrt{3}$, where r_{LM} is the sensor radius used in lawnmower path construction.

We will test two lawnmower algorithms. In algorithm Lawnmower 1, $r_{LM} = 0.5D_{\max}(1 - 0.5\epsilon)$ is set to be the mean sensor radius when the terrain has 100% uncertainty. In algorithm Lawnmower 2, $r_{LM} = 0.25D_{\max}(1 - 0.5\epsilon)$. In our search scenario, the stopping time of the algorithm is the time when the sensor finds the target. If we run a large number M of search simulations for Lawnmower 1 and MAS algorithms on the terrain given in Fig. 16, for any given search precision p one can calculate, based on the statistics, the smallest stopping times T needed to achieve it. That is, the number of searches that successfully found the target by time T divided by M equals p. Now, for any given search precision, we can compare the corresponding stopping times for Lawnmower 1 and MAS search strategies. In Fig. 17, we plotted the ratio of stopping times as a function of search precision for Lawnmower 1 and MAS for uncertainty ϵ in terrain 20%, 50%, and 100%, respectively. From the plot, one can see that at very low and very high precision levels, the MAS algorithm finds the target faster than the lawnmower algorithm. If we intersect light grey, dark grey, and black curves on Fig. 17 with the horizontal line at 1, we can see that there is a precision interval in which the lawnmower algorithm finds the target faster, this means that if we knew the uncertainty in terrain beforehand, it would be possible to design a lawnmower algorithm that would find the target faster than

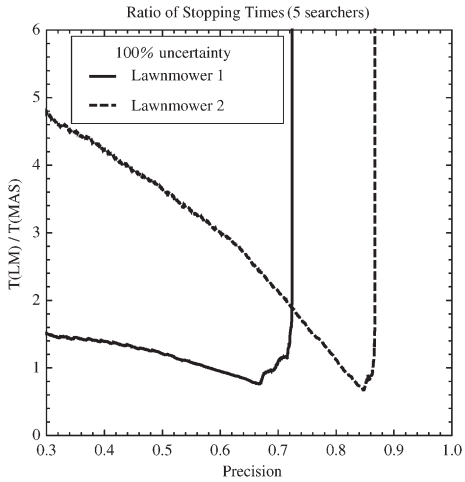

Fig. 18. Ratio of the search stopping times for lawnmower and MAS strategies as a function of the search precision (probability of the search success at stopping) for different designs of lawnmower trajectories.

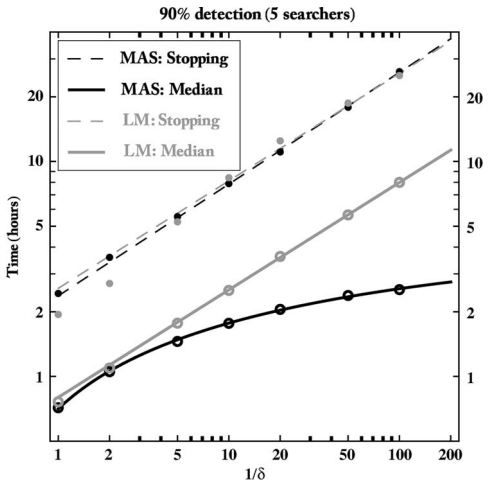

Fig. 19. Median search time (solid curves) and search stopping time (dashed curves) for lawnmower (grey) and MAS (black) strategies as a function of a measure of terrain uncertainty. Median search time of the MAS search converges logarithmically with the measure of uncertainty.

the MAS algorithm for a given precision level. However, as the precision level gets higher, the MAS algorithm will ultimately beat the lawnmower. In Fig. 17, this fact is illustrated by the steep right end of each curve with vertical asymptote. This means that any lawnmower algorithm has an upper limit to its precision level. The limit in precision level in lawnmower comes from the fact that for the pre-computed lawnmower path there always will be targets that the lawnmower will never find. In Fig. 18, solid line shows the ratio of the search stopping times for MAS search (executed with 5 sensors and 100% uncertainty in terrain) and Lawnmower 1; dashed line shows the ratio of the search stopping times for MAS search (executed with five sensors and 100% uncertainty in terrain) and Lawnmower 2. The ratio of the search stopping times for five sensors (solid curve in Fig. 18) is very similar to the ratio of the search stopping times for one sensor (light gray curve in Fig. 17) which indicates that the MAS search time is scalable with the number of sensors. As shown in Fig. 18, the Lawnmower 2 coverage, that is four times as dense as Lawnmower 1, does not have better stopping time than Lawnmower 1 when the precision is less than 0.73. For precisions between 0.82 and 0.84, Lawnmower 2 performs better than the MAS algorithm. The minimum of the ratio of stopping times of Lawnmower 2 and MAS search is about the same as the ratio of stopping times of Lawnmower 1 and MAS search, which is about 0.7. This illustrates that higher precision can be achieved by choosing a denser Lawnmower path, however the same choice of path will perform very poorly if we decide that lower precision is enough.

The MAS search algorithm is universal in the sense that its trajectories will always achieve a required precision and it will outperform the lawnmower algorithm for every precision except for a very small interval of precision where the lawnmower is a little bit faster. This shows that lawnmowerlike algorithms with pre-computed paths may be a very poor choice of coverage algorithm if the terrain is uncertain, as they may not be able to reach the required precision at all. The MAS search on the other hand will always be able to achieve the desired precision with stopping times that will be most likely much better than those of the lawnmower.

Finally, let us compare the performance of MAS and lawnmower algorithms as a function of the measure of terrain uncertainty $1/\delta$. Here, $\delta = (D_{\min}/D_{\max})^2 = (1-\epsilon)^2$ is the fraction of the sensor footprint area that is detectable. As we found earlier in this section, for a given precision (say 90%), we can design a lawnmower algorithm that will have the same stopping time as the MAS algorithm (design the best lawnmower). Fig. 19 shows a log-log plot of the median search time of MAS search (solid black curve) and median search time of the best lawnmower search (solid grey curve) as a function of the measure of terrain uncertainty. We found that in order to match the MAS stopping time, the lawnmower parameter r_{LM} has to be chosen as $r_{LM} = 0.5 D_{\max}\delta^{0.26}$. Fig. 19 shows that lawnmower stopping time is almost matched to the MAS stopping time. Median search time of the MAS search grows logarithmically ($\sim\log(1/\delta)$) whereas the search time of lawnmower grows as a square root ($\sim\sqrt{1/\delta}$) of the measure of uncertainty. This, again, shows the advantage of MAS search as compared to the best lawnmower design when the terrain is uncertain.

VII. Conclusion

We have presented the MAS algorithm that can be implemented on a domain with any given geometry and finds a target with probability of detection of at least P_D and probability of false alarm of at most P_{FA}. The domain may also contain foliage that the sensors cannot penetrate. The MAS search algorithm features a novel two-step decision-making algorithm that is based on the Neyman–Pearson lemma and realistic second-order dynamics of sensors. We use the SMC algorithm described in [21] to compute sensor trajectories that provide uniform coverage of the domain. By uniform coverage, we roughly mean that points on the sensor trajectories must be uniformly distributed or evenly spaced throughout the domain. The MAS search algorithm is fairly easy to implement and has computational complexity $O(n\log(n))$. Because the SMC coverage is expected to distribute the sensors uniformly

throughout the domain at any given point in time, the MAS search algorithm is very robust to the uncertainty in sensor radius. In particular, MAS search outperforms lawnmower-type search where the trajectories are computed in advance, provided that the distribution of sensor radius is given. An important feature of MAS search algorithm is that it is capable of computing the sensor trajectories for virtually any geometry of the search domain. The MAS search algorithm in presence of uncertainty vastly outperforms random search (Billiard search) as shown in Table II. We conjecture that in contrast with random search the MAS algorithm maximizes its effective use of assets: The median search time is inversely proportional to the number of sensors.

APPENDIX
SPECTRAL MULTISCALE COVERAGE

There are n sensors and their dynamics is described by

$$\ddot{x}_j(t) = -c\dot{x}_j(t) + u_j(t) \tag{16}$$

where $c > 0$ is a damping coefficient to model the resistance due to air or water, and $u_j(t)$ is the control (or force) applied on each moving sensor. Given trajectories $x_j : [0, t] \to \mathbb{R}^n$, for $j = 1, 2, \ldots, n$, we define the distribution C^t as

$$C^t(x) = \frac{1}{Nt} \sum_{j=1}^{N} \int_0^t \delta\left(x - x_j(\tau)\right) d\tau. \tag{17}$$

$\delta(.)$ is the Dirac delta distribution. The inner product of C^t with a bounded function f is given as

$$\langle C^t, f \rangle = \frac{1}{Nt} \sum_{j=1}^{N} \int_0^t f\left(x_j(\tau)\right) d\tau = \frac{1}{Nt} \int_0^t \sum_{j=1}^{N} f\left(x_j(\tau)\right) d\tau. \tag{18}$$

Let f_k be the Fourier basis functions that satisfy Neumann boundary conditions on a rectangular domain U and k is the corresponding wave-number vector. For instance, on a rectangular domain $U = [0, L_1] \times [0, L_2]$, we have

$$f_k(x) = \frac{1}{h_k} cos(k_1 x_1) cos(k_2 x_2), \text{ where}$$

$$k_1 = \frac{K_1 \pi}{L_1} \text{ and } k_2 = \frac{K_2 \pi}{L_2},$$

$$\text{for } K_1,\ K_2 = 0, 1, 2 \ldots . \text{ and where}$$

$$h_k = \left(\int_0^{L_1} \int_0^{L_2} cos^2(k_1 x_1) cos^2(k_2 x_2) dx_1 dx_2 \right)^{1/2}. \tag{19}$$

The division by the factor h_k ensures that f_k has L^2 norm equal to one. Therefore, f_k is an orthonormal basis. Now, computing the Fourier coefficients of the distribution C^t, we have

$$c_k(t) = \langle C^t, f_k \rangle = \frac{1}{Nt} \sum_{j=1}^{N} \int_0^t f_k\left(x_j(\tau)\right) d\tau. \tag{20}$$

The Fourier coefficients of the prior probability distribution $\mathcal{P}$ are given as

$$\mu_k = \langle \mathcal{P}, f_k \rangle. \tag{21}$$

The metric for ergodicity (or uniformity of coverage) we use is given by a Sobolev space norm of negative index (H^{-s}, for $s = (d+1)/2$, $d =$ dimension of the space). i.e.,

$$\phi^2(t) = \|C^t - \mathcal{P}\|_{H^{-s}}^2 = \sum_K \Lambda_k \left| s_k(t) \right|^2,$$
$$\text{where } s_k(t) = c_k(t) - \mu_k \text{ and } \Lambda_k = \frac{1}{(1 + \|k\|^2)^s}. \tag{22}$$

Requiring $\phi^2(t)$ to converge to zero is the same as requiring time averages of the Fourier basis functions functions along trajectories to converge to the spatial averages of the basis functions. In other words, $\phi^2(t)$ quantifies how much the time averages of the Fourier basis functions deviate from their spatial averages, but with more importance given to large-scale modes than the small-scale modes. For more justification on the use of this as a metric for uniform coverage, see [21] and [28].

The objective of the uniform coverage algorithm is to find feedback laws so as to drive the metric $\phi^2(t)$ to zero. Let $u_{\max}$ be the maximum force that can be applied on the moving sensor. It can be shown that the control $u_j(t)$ that makes the second time derivative of $\phi^2(t)$ as negative as possible is given as

$$u_j(t) = -u_{\max} \frac{B_j(t)}{\|B_j(t)\|_2},$$
$$\text{where } B_j(t) = \left[\sum_k \Lambda_k s_k(t) \nabla f_k\left(x_j(t)\right) \right]$$
$$\text{and } \Lambda_k = \frac{1}{(1 + \|k\|^2)^s}. \tag{23}$$

This feedback control law can also be shown to be the optimal control that minimizes the uniform coverage metric at the end of an infinitesimal time horizon. Note that the feedback law is in terms of the Fourier coefficients $c_k(t)$. Therefore, the algorithm requires the constant updating of integrals of the type

$$C_k(t) = \sum_{j=1}^{n} \int_0^t f_k\left(x_j(\tau)\right) d\tau \tag{24}$$

for all wave-numbers K.

REFERENCES

[1] J. R. Frost and L. D. Stone, Review of search theory: Advances and applications to search and rescue decision support, U.S. Coast Guard Res. Develop. Center, New London, CT, Tech. Rep. CG-D-15-01. [Online]. Available: http://handle.dtic.mil/100.2/ADA397065

[2] D. Blackwell, "Discrete dynamic programming," *Ann. Math. Stat.*, vol. 33, no. 2, pp. 719–726, Jun. 1962.

[3] N.-O. Song and D. Teneketzis, "Discrete search with multiple sensors," *Math. Methods Oper. Res.*, vol. 60, no. 1, pp. 1–13, Sep. 2004.
[4] C. Cai and S. Ferrari, "Information-driven sensor path planning by approximate cell decomposition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 3, pp. 672–689, Jun. 2009.
[5] Y. Zhao, S. D. Patek, and P. A. Beling, "Decentralized bayesian search using approximate dynamic programming methods," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 970–975, Aug. 2008.
[6] R. R. Brooks, J. Schwier, and C. Griffin, "Markovian search games in heterogeneous spaces," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 3, pp. 626–635, Jun. 2009.
[7] X.-F. Xie and J. Liu, "Multiagent optimization system for solving the traveling salesman problem (TSP)," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 489–502, Apr. 2009.
[8] S.-P. Hong, S.-J. Choa, and M.-J. Park, "A pseudo-polynomial heuristic for path-constrained discrete-time Markovian-target search," *Eur. J. Oper. Res.*, vol. 193, no. 2, pp. 351–364, Mar. 2009.
[9] D. E. Knuth, *The Art of Computer Programming*, vol. 3, 2nd ed. Reading, MA: Addison-Wesley Professional, 1998.
[10] L. D. Stone, *Theory of Optimal Search*. New York: Academic, 1975.
[11] S. Ferrari, R. Fierro, B. Perteet, C. Cai, and K. Baumgartner, "A geometric optimization approach to detecting and intercepting dynamic targets using a mobile sensor network," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 292–320, Feb. 2009.
[12] T. Wettergren and J. Baylog, "Collaborative search planning for multiple vehicles in nonhomogeneous environments," in *Proc. OCEANS*, Biloxi, MS, 2009, pp. 1–7.
[13] J. Riehl, G. E. Collins, and J. Hespanha, "Cooperative graph-based model predictive search," in *Proc. 46th IEEE Conf. Decision Control*, New Orleans, LA, Dec. 2007, pp. 2998–3004.
[14] B. DasGupta, J. Hespanha, J. Riehl, and E. Sontag, "Honeypot constrained searching with local sensory information," *Nonlinear Anal., Theor. Methods Appl.*, vol. 65, no. 9, pp. 1773–1793, Nov. 2006.
[15] P. Vincent and I. Rubin, "A framework and analysis for cooperative search using UAV swarms," in *Proc. ACM Symp. Appl. Comput.*, Nicosia, Cyprus, Mar. 2004, pp. 79–86.
[16] Y. Jin, Y. Liao, A. A. Minai, and M. M. Polycarpou, "Balancing search and target response in cooperative unmanned aerial vehicle (UAV) teams," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 3, pp. 571–587, Jun. 2006.
[17] Y. Yang, M. Polycarpou, and A. Minai, "Opportunistically cooperative neural learning in mobile agents," in *Proc. Int. Joint Conf. Neural Netw.*, Honolulu, HI, May 2002, pp. 2638–2643.
[18] K. Baumgartner and S. Ferrari, "A geometric transversal approach to analyzing track coverage in sensor networks," *IEEE Trans. Comput.*, vol. 57, no. 8, pp. 1113–1128, Aug. 2008.
[19] J. S. T. Wettergren and R. Streit, "Tracking with distributive sets of proximity sensors using geometric invariants," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 4, pp. 1366–1374, Oct. 2004.
[20] K. Baumgartner, S. Ferrari, and T. Wettergren, "Robust deployment of dynamic sensor networks for cooperative track detection," *IEEE Sensors J.*, vol. 9, no. 9, pp. 1029–1048, Sep. 2009.
[21] G. Mathew and I. Mezić, "Spectral multiscale coverage: A uniform coverage algorithm for mobile sensor networks," in *Proc. 48th IEEE Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 7872–7877.
[22] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," *Philos. Trans. Roy. Soc. London A, Math. Phys. Eng. Sci.*, vol. 231, pp. 289–337, 1933.
[23] D. Chudnovsky and G. Chudnovsky, "Search plans generated by billiards. Search theory," in *Lecture Notes in Pure and Applied Mathematics*, vol. 112. New York: Dekker, 1989, pp. 103–130.
[24] R. J. Palmer, "Guiding principles in controlling small automated vehicles," in *Proc. IEEE WESCANEX. Commun., Power, Comput. Conf.*, Winnipeg, MB, Canada, May 1995, pp. 366–370.
[25] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[26] B. O. Koopman, "Search and its optimization," *Amer. Math. Monthly*, vol. 86, no. 7, pp. 527–540, Aug./Sep. 1979.
[27] J. L. Solka, D. J. Marchette, B. C. Wallet, V. L. Irwin, and G. W. Rogers, "Identification of man-made regions in unmanned aerial vehicle imagery and videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 852–857, Aug. 1998.
[28] G. Mathew and I. Mezić. (2011, Feb.). Metrics for ergodicity and design of ergodic dynamics for multi-agent systems. *Phys. D: Nonlinear Phenom.* [Online]. *240(4/5)*, pp. 432–442. Available: http://dx.doi.org/10.1016/j.physd.2010.10.010 2010

Alice Hubenko received the M.S. degree in mathematics from Eötvös University, Budapest, Hungary, in 1998, and the Ph.D. degree in mathematics from the University of Memphis, Memphis, TN, in 2004.

During 2005–2006, she was a Lecturer in the Department of Mathematics, University of California, Riverside. Since 2007, she has been a Postdoctoral Fellow in the Department of Mechanical Engineering, University of California, Santa Barbara. Her research interests include graph theory and combinatorics, algorithms, and complex networks.

Vladimir A. Fonoberov received the M.S. degree in physics and the Ph.D. degree in theoretical/computational physics from Moldova State University, Moldova, in 1999 and 2002, respectively.

He was a Postdoctoral Fellow in the Department of Electrical Engineering, University of California, Riverside, during 2003–2006. Since 2007, he has been with AIMdyn, Inc., Santa Barbara, California, where he is currently a Chief Scientist. He and his team at AIMdyn, Inc. developed mathematical and physical models of complex interconnected systems as well as novel tools to analyze model robustness and accelerate model analysis. He has published 30 articles in refereed journals and has over 900 citations. He works in the fields of modeling of physical systems and uncertainty management and their applications including molecular and fluid dynamics, uncertainty quantification in mathematical and physical models, agent-based modeling, parallel programming and software development.

Dr. Fonoberov won an NSF Postdoctoral Fellowship Award in Science and Engineering in 2003. He also received an Inventor Recognition Award from Semiconductor Research Corporation in 2009.

George Mathew was born in Cincinnati, OH, on November 22, 1978. He received the B.Tech degree in mechanical engineering from the University of Kerala, Trivandrum, India, in 2000, and the Ph.D. degree in mechanical engineering from the University of California, Santa Barbara, in 2006.

During 2007–2008, he was a Senior Scientist in the Dynamics and Optimization Group at the United Technologies Research Center (UTRC), East Hartford, CT. During 2008–2010, he was a Postdoctoral Scholar at the University of California, Santa Barbara. Since 2010, he has been a Senior Scientist with the Embedded Systems and Networks Group, UTRC Inc., Berkeley, CA. His research interests include dynamical systems, optimal control, control of fluid mixing, mobile sensor networks and stochastic hybrid systems.

Igor Mezić (M'08) received the Dipl. Ing. degree in mechanical engineering from the University of Rijeka, Rijeka, Croatia, in 1990, and the Ph.D. degree in applied mechanics from the California Institute of Technology, Pasadena.

He was a Postdoctoral Researcher at the Mathematics Institute, University of Warwick, Coventry, U.K., in 1994–1995. From 1995 to 1999, he was a member of the Mechanical Engineering Department, University of California, Santa Barbara, where he is currently a Professor. In 2000–2001, he was an Associate Professor in the Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA. He is the Director of the Center for Energy Efficient Design and Head of the Buildings and Design Solutions Group at the Institute for Energy Efficiency at the University of California, Santa Barbara. He works in the field of dynamical systems and control theory and applications to complex systems.

Dr. Mezić is an Editor of *Physica D: Nonlinear Phenomena* and an Associate Editor of the *Journal of Applied Mechanics* and *SIAM Journal on Control and Optimization*. He won the Alfred P. Sloan Fellowship, an NSF CAREER Award, and the George S. Axelby Outstanding Paper Award on "Control of Mixing" from the IEEE. He also won the United Technologies Senior Vice President for Science and Technology Special Achievement Prize in 2007.