The Cost of Reality: Effects of Real-World Factors on Multi-Robot Search

Jim Pugh and Alcherio Martinoli

Abstract— **Designing algorithms for multi-robot systems can be a complex and difficult process: the cost of such systems can be very high, collecting experimental data can be time-consuming, and individual robots may malfunction, invalidating experiments. These constraints make it very tempting to work using high-level abstractions of the robots and their environment. While these high-level models can be useful for initial design, it is important to verify techniques in more realistic scenarios that include real-world effects that may have been ignored in the abstractions. In this paper, we take a simple, coordinated, multi-robot search algorithm and illustrate problems that it encounters in environments which incorporate real-world factors, such as probabilistic target detection and positional noise. We compare the performance to that of several simple randomized approaches, which are better able to deal with these constraints.**

I. Introduction

Locating one or more targets within an unknown environment is a task well-suited to mobile robotics. Robots can be equipped with sensors to detect targets and programmed to explore the area in search of their goal(s). The automated nature of this approach may save much time and effort as compared to other search methods. Performance may be further improved by use of multiple robots in the process, which will decrease the time needed to complete the search task and increase robustness to failures of individual robots. Robotic search is especially preferable when the area is either hazardous or inaccessible to humans. Examples include locating mines for de-mining [8], [1], finding victims in a disaster area [11], and planetary exploration [13]. Although search has been well-explored in the past [2], using multi-robot systems for search is a more recent development and has not yet been studied extensively.

Algorithms for multi-robot systems can be roughly classified in a spectrum based on the level of coordination involved. At one end of this spectrum are completely coordinated algorithms; these techniques use strict rules to direct the behavior of the individual robots, typically assigning each robot a specific task in order to maximize the efficiency. At the other end of the spectrum are completely randomized algorithms; these minimize explicit control of the robot's actions and instead behave in a directed random fashion, with the expectation that the task goals will eventually be completed. While coordinated approaches tend to accomplish their objectives more quickly than randomized approaches, randomized approaches are often easier to design and may

be more robust to disruptions due to their simplicity. Many algorithms lie somewhere in the middle of the spectrum, incorporating both coordinated and randomized aspects in order to (hopefully) exploit the best of both worlds.

Because of the cost and effort required to run experiments using real multi-robot systems, it is often desirable to design algorithms using a simulation or model of the actual robotic group. This allows for fast and easy measurement of the performances of different techniques. The potential problem with this approach is that often these abstractions do not incorporate many factors which may be present in a real-world scenario. Examples might include sensor noise, wheel slip, and limited communication ability. Depending upon the algorithm, these factors may have a significant impact on the system performance. If design is only done using these high-level abstractions without verification in a more realistic environment, techniques may be developed which have very poor real-world performance. This is especially true of highly coordinated algorithms, where the complexity provides more opportunities for things to go wrong.

Although it has long been believed that simple, randomized algorithms are more robust to malfunctions and unexpected factors than highly coordinated ones, there has thus far been few experiments which clearly exhibit this effect. The impact of introducing real-world aspects in multi-robot search was partially covered by Douglas W. Gage [7], with an analysis of coordinated approaches versus randomized ones in the presence of probabilistic target detection. This paper aims to expand upon that work and demonstrate some potential pitfalls of coordinated multi-robot search approaches and to show that randomized algorithms are able to overcome these challenges.

Section II introduces our techniques, analyzes their expected performances, and simulates them in a mostly noise-less environment. In Section III, we observe how sensor noise affects the different algorithms. Section IV studies how uncertainty in position impacts the performance. In Sections V and VI, we discuss the implications of our results and conclude.

II. Analysis and Simulation in Ideal Environments

In order to locate a search target in an environment, robots must move about the environment until they are able to detect the presence of the target. Detection typically occurs when a robot is within some close proximity to the target's location. Depending on the type of target, additional searching may be necessary after the initial detection in order to precisely ascertain its location (e.g., following an odor plume to an

odor source). For our case study, we consider detection to be equivalent to localization (such as would be the case with a visual detection).

We focus here only on the localization of immobile targets; this is the case for many search tasks (e.g., demining, finding wounded victims in a disaster area). Effective techniques for this type of search can be fundamentally different than those used for tracking of multiple mobile targets (see for example [15]) where robots typically must monitor the locations of the targets over time instead of only obtaining a single detection. If targets are immobile, the most effective method of searching is systematically sweeping unvisited areas of the environment until all targets are discovered.

In high-level abstractions of search scenarios, target detectors are often modeled as fixed-range perfect sensors. A popular approach to modeling search is by using "approximate cellular decomposition" where the environment is described by a discrete grid of cells (as described in [6] and employed in [5] and [16]). In these scenarios, robots are capable of detecting all targets located in some fixed configuration of nearby cells. However, real-world search rarely occurs in discrete space, and in order to more closely model reality, we will confine our analysis and simulations to continuous space. In continuous space, a fixed-range perfect sensor might be able to always detect any target within some radius r of its position, and we will use this definition for our analysis.

A. Coordinated Multi-Robot Search

In a fully coordinated multi-robot approach to a coverage/search task, the environment is typically divided up into distinct regions of equal area, and each robot is responsible for covering a different region (for example, in [3], [12], and [18]). In this way, there will be minimal overlap between sensor coverage of the robots, which will maximize the area covered over time and therefore minimize the time needed for complete coverage. If we also assume that robots cover their area such that they never sweep any area more than once, the fraction of the environment covered over time, $C(t)$, can be expressed as:

$$C(t) = \frac{2rvNt}{A}$$

where v is the constant velocity of the robots, N is the number of robots, and A is the total area in which the target may be located. Because this expression is deterministic, we can solve for the time needed to completely cover the environment by setting $C(t_{c,max}) = 1$:

$$t_{c,max} = \frac{A}{2rvN}$$

Because detection is perfect and every point in the environment has been swept by time $t_{c,max}$, we are guaranteed to have detected all targets by this time. If we assume all targets are distributed independently with uniform probability, the chance of having detected all targets by time t can be expressed as:

$$\begin{aligned} P_c(t) &= C(t)^M \\ &= (\frac{2rvNt}{A})^M \end{aligned}$$

where $P_c(t)$ is the cumulative distribution function and M is the number of targets. Using this, we can take the derivative to get the probability distribution function and calculate the expected time to completion:

$$\begin{aligned} E[t_{c,end}] &= \int_0^{t_{c,max}} t p_c(t) dt \\ &= \frac{M}{M+1} \frac{A}{2rvN} (\frac{2rvNt_{c,max}}{A})^{(M+1)} \\ &= \frac{M}{M+1} t_{c,max} \end{aligned}$$

In the case of a single target ($M = 1$), we get $E[t_{c,end}] = \frac{t_{c,max}}{2}$.

B. Random Multi-Robot Search

In a randomized multi-robot approach to a search task, robots will move about the environment without any explicit plan for where they are going. While this is easier to implement than the coordinated approach, it quickly becomes very likely that the same or different robots will sweep areas of the environment multiple times, due to the lack of navigational skills and inter-robot coordination, respectively. With perfect sensors, covering an area more than once is a waste of time, and therefore the search will typically take longer to complete. Also, because there exists no plan to explicitly explore all areas in the environment, there is no finite bound on the time it will take to achieve complete coverage, and hence no maximum time to completion of search.

For a single robot, we can express the probability of detecting a single target over some small time step Δt as:

$$p_{\Delta t} = \frac{2rv\Delta t}{A}$$

if we assume that Δt is small enough that the robot will not cover the same area more than once in that period. The cumulative probability over some time period t is therefore:

$$P_r(t, \Delta t) = 1 - (1 - \frac{2rv\Delta t}{A})^{\frac{t}{\Delta t}}$$

Taking the limit as Δt goes to 0, we get a cumulative distribution function of:

$$P_r(t) = 1 - e^{\frac{-2rvt}{A}}$$

In the case of M different targets, the probability of detecting all targets is:

$$P_r(t) = (1 - e^{\frac{-2rvt}{A}})^M$$

We can take the derivative to get the probability density function and calculate the expected completion time as:

$$
\begin{aligned}
E[t_{r,end}] &= \int_0^\infty tp_r(t)dt \\
&= tP_r(t)|_0^\infty - \int_0^\infty P_r(t)dt \\
&= \frac{A}{2rv}\sum_{i=1}^M \frac{1}{i}
\end{aligned}
$$

In the case of multiple robots, the performance depends on how they interact while searching. If N different robots are all exploring different parts of the environment, we can assume the sensor coverage is effectively multiplied by N. This could be achieved with inter-robot repulsion by allowing robots to detect the relative location of nearby other robots, which can be accomplished using a simple on-board relative positioning system [17]. The expected value then becomes:

$$
E[t_{rr,end}] = \frac{A}{2rvN}\sum_{i=1}^M \frac{1}{i}
$$

The expected time to completion therefore decreases as $\frac{1}{N}$ as the number of robots increase, just as it does in the coordinated algorithm. For the case of $M = 1$, or a single target, the expected time would be $\frac{A}{2rvN}$ or twice that of the coordinated search.

If robots move independently of one another and may have overlapping coverage, this will decrease the average coverage at any time. Let us assume that amount of non-overlapping coverage for a robot is dependent upon the proximity of the nearest other robot and is given by:

$$
c_{\Delta t}(d) = \frac{(0.5d + r)v\Delta t}{A}
$$

where d is the distance to the nearest robot and $d < 2r$. This means that at a range of $2r$, coverage is full, while at a range of 0, coverage is halved because of complete overlap. Assuming a uniformly random independent distribution of robots in the environment, the probability of a robot having a neighbor robot within range d is:

$$
P(d) = \frac{\pi d^2}{A}
$$

and therefore the probability that any of $N - 1$ other robots are within distance d is:

$$
P(d) = 1 - (1 - \frac{\pi d^2}{A})^{N-1}
$$

If we assume that the detection area is much smaller than the environment size, we can approximate $P(d)$ as:

$$
P(d) = \frac{(N-1)\pi d^2}{A}
$$

and therefore:

$$
p(d) = \frac{2(N-1)\pi d}{A}
$$

We can use this to calculate the expected coverage of a robot by combining the expected diminished coverage with the expected non-overlapping coverage:

$$
\begin{aligned}
E[p_{\Delta t}] &= \int_0^{2r} c_{\Delta t}(x)p(x)dx + c_{\Delta t}(2r)(1 - P(2r)) \\
&= \frac{20(N-1)\pi v\Delta t r^3}{3A^2} + \frac{2rv\Delta t}{A}(1 - \frac{4(N-1)\pi r^2}{A}) \\
&= \frac{2rv\Delta t}{A}(1 - \frac{2(N-1)\pi r^2}{3A})
\end{aligned}
$$

This is the non-overlapping coverage minus a penalty that grows linearly with the number of robots. Let us denote:

$$
\eta = \frac{2(N-1)\pi r^2}{3A}
$$

Then the expected completion time becomes:

$$
E[t_{r,end}] = \frac{A}{2r(1-\eta)vN}\sum_{i=1}^M \frac{1}{i}
$$

This results in a larger expected completion time than for the randomized algorithm without sensor overlap. However, if the area of the environment is much larger than the detection radius, the penalty will not have a major effect.

C. Simulation

To observe how these techniques actually perform, we simulate them using the realistic simulator Webots [14]. For our case study, we take the most simple scenario: a single target within a rectangular arena, with robots having prior knowledge about the geometry. The arena is 8.0m x 8.0m, and we use simulations of the e-puck[1] robot [4] as our searching robot. We allow robots to sense targets that are within 20 cm of their position. We use three techniques which closely match those described above:

Coordinated: Robots initially begin arranged with left-aligned equal spacing along the bottom of the arena. Each robot follows a pre-prescribed path which takes it straight up to the top of the arena, followed by a short step to the right, then back down to the bottom, another short step right, and repeats (see Fig. 1). In this way, robots cut swaths through the unexplored space with their detection sensors, and they can deliberatively cover the entire arena. Path following is accomplished by robots always moving directly towards a current target point until it is reached (within 2 cm), then continuing on to the next one. The space in the arena is divided to give an equal portion to each robot present. If a robot finishes sweeping its area without finding the target (due to unreliable target detection), it will backtrack along its path to sweep the same area again and repeat the process until the target is found.

Random: Robots are initially placed randomly within the arena. Throughout the experiment, they continually try to move forward, turning to avoid any obstacles (i.e. walls and robots) they may encounter. In order to increase randomness, Gaussian noise is added to the speed values given to each
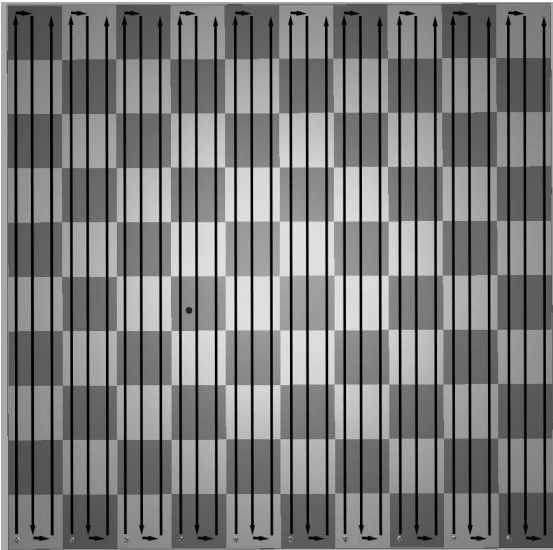
[1]http://www.e-puck.org

Fig. 1. Paths of 10 Robots in Coordinated Approach

Fig. 2. Search performance of different techniques over 100 runs; asterixes represent analytical expected values and error bars represent 95% confidence intervals assuming an Exponential distribution

wheel at every time step. This should theoretically cause them to move to any region in the arena with approximately equal probability.

Random-Repulse: The same setup and technique as Randomized, but robots use relative positioning to create a potential field repulsion between one another, which causes them to rarely to come within overlapping detection range of each other.

Because the detection area of the robots is circular and the arena is rectangular, if the robots use a swath width of the full detection range $2r$ with the back-and-forth sweeping pattern, there will be small detection gaps not covered by the sensors along the arena bounds. Initial simulations showed that using this techniques caused the robots to never find the target on some rare occasions. In order to remedy this, we inscribe a square within the detection circle and use this sensor geometry for our path planning. This results in swath widths of $r\sqrt{2}$, which covers the entire arena with some overlap, but increases the path length by a factor of $\sqrt{2}$ on average, which should increase the expected coverage time by the same factor. While it would be possible to completely cover the arena more quickly using other approaches, these require heuristic changes which may bias the analysis.

For a single search experiment, a target was placed randomly at some location within the arena. Robots move at 6.44 cm/s and continue searching until the target is found, or until a timeout of 128,000 seconds has passed.

D. Results

A bar chart of the average search time for the different techniques over 100 runs using different numbers of robots can be seen in Fig. 2. All techniques completed the search before timeout in all scenarios. We can see that Coordinated outperforms both Random approaches in all cases, though not by a significant margin. There is no clear distinction in performance between Random and Random-Repulse.
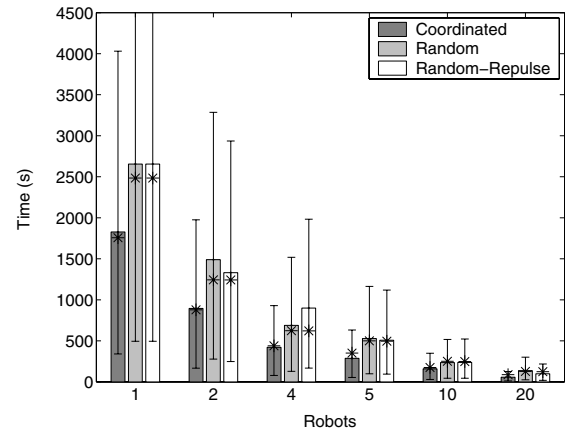
We can compare the results obtained with what we calculate analytically. The expected time for each technique with different numbers of robots can be seen in Fig. 2 as asterixes overlaid upon the bars. The expected time for the Coordinated search matched very closely to the simulated time. The simulated times for Random search, though close, were slightly higher than expected. This suggests that the robots' behavior for the Random techniques may not actually be entirely random, and certain areas of the arena may be covered less often than others, resulting in higher search times for certain target positions. This issue was previously discussed in [7], and several potential solutions were suggested, but the implementation of them is beyond the scope of this work. The difference in performance between Random and Random-Repulse techniques was not significant in either simulation or expected completion time; this is due to the very small value of η in these scenarios.

III. NOISY DETECTION OF TARGETS

We have thus far assumed that robots have perfect detection of targets within a fixed radius. In reality, this is not a realistic assumption. Most sensors have a probabilistic response, where the likelihood of correctly detecting a target decreases as the distance to the target increases. Using a fixed radius perfect sensor is a rough approximation of this response, since it does limit the detection range, but possible anomalies due to this approximation should not be ignored. We therefore compare our techniques using noisy detection of targets.

There exists two types of incorrect detections due to sensor noise: false negatives, where a target is within sensor range but none is detected, and false positives, where no target is within range but one is incorrectly detected. We ignore the case of false positives here, as we assume that any target detection will be immediately verified after (e.g., the robot could more closely examine the area to double-check the detection), and focus our efforts on false negatives.

For many types of detection (e.g., sound, radio), the important value to consider for detection is the signal-to-

noise ratio (SNR) of target emission to background noise. The probability for detection might be reflected by:

$$p_{det} = \frac{P_s}{P_s + N_0}$$

where P_s is the received signal power, and N_0 is the background noise, which we assume to be uncorrelated with the received signal. P_s would be determined by:

$$P_s(d) = \frac{P_0}{d^2}$$

where d is the distance between the robot and the target. The full expression for the detection probability would then be:

$$
\begin{aligned}
p_{det}(d) &= \frac{\frac{P_0}{d^2}}{\frac{P_0}{d^2} + N_0} \\
&= \frac{1}{1 + \frac{N_0}{P_0} d^2} \\
&= \frac{1}{1 + (\frac{d}{d_h})^2}
\end{aligned}
$$

where $d_h = \sqrt{P_0/N_0}$ is the distance at which the probability of detection becomes 0.5, which we denote as the half-power distance. We also assume that we have some maximum detection range, d_{max}, and targets farther than this distance have zero probability of being detected by the robot, since even though our model might give a small probability of detection, it is unrealistic to "locate" a target which is far from the robot.

If we assume that robots move approximately straight most of the time, we can calculate the probability of detection of a target based on its lateral displacement from the robot as it passes. Targets with lateral displacement greater than d_{max} have no chance of being detected. Targets with very small lateral displacement are in sensor range for more time and have a larger probability of being detected since they are closer to the robot. The probability of detecting a target with lateral displacement x and front/back displacement y is:

$$p_{det}(x, y) = \frac{1}{1 + (\frac{x}{d_h})^2 + (\frac{y}{d_h})^2}$$

for $\sqrt{x^2 + y^2} \leq d_{max}$. Let us also assume that the probability of detection given is the probability of detecting a target over one time step, and if we shrink the size of the time step, we get:

$$p_{det}(x, t_i, t_f) = 1 - \prod_{t=t_i/\Delta t}^{t_f/\Delta t} (1 - p_{det}(x, y(t))^{\Delta t}$$

for the probability of detecting from time t_i to t_f. Taking the limit as Δt goes to zero, we get:

$$p_{det}(x, t_i, t_f) = 1 - \exp\left(\int_{t_i}^{t_f} \ln(1 - p_{det}(x, y(t))) dt\right)$$

If we assume constant velocity v to get $y(t) = vt$ and input the time bounds of $t_i = -\sqrt{d_{max}^2 - x^2}/v$ and $t_f =$ $\sqrt{d_{max}^2 - x^2}/v$ for the period when the robot will be within detectable range of the target, the equation becomes:

$$p_{det}(x) = 1 - \exp\left(\int_{-\sqrt{d_{max}^2-x^2}/v}^{\sqrt{d_{max}^2-x^2}/v} \ln(1 - p_{det}(x, vt)) dt\right)$$

For the Coordinated algorithm, robots always follow the same path, so if a target is on the path periphery with a low probability of detection, it will have that same low probability every time the robot repeats the search. The expected number of misses for a target can be calculated based on its lateral detection probability as:

$$m(x) = \frac{1 - p_{det}(x)}{p_{det}(x)}$$

If we assume that the target has uniform probability of assuming any lateral displacement along any robot's sweep path, we can calculate the overall expected number of missed passes m by:

$$m = \frac{1}{2r_s} \int_{-r_s}^{r_s} \frac{1 - p_{det}(x)}{p_{det}(x)} dx$$

where r_s is the detection sweep radius, and the expected time then becomes the lost time from missed passes plus the original expected time:

$$E[t_{c,end}] = m t_{max} + \frac{t_{max}}{2} = (m + \frac{1}{2}) t_{max}$$

For randomized search, because robots do not follow any path, the probability of detection is independent of the target position. Instead, if we also assume that robots move straight most of the time, we only need to adjust the detection probability by a multiple ρ given by:

$$\rho = \frac{1}{2d_{max}} \int_{-d_{max}}^{d_{max}} p_{det}(x) dx$$

which is the probability of detecting a target within the sensor coverage area, and the expected completion times then become:

$$
\begin{aligned}
E[t_{r,end}] &= \frac{A}{2r\rho(1 - \eta)vN} \sum_{i=1}^{M} \frac{1}{i} \\
E[t_{rr,end}] &= \frac{A}{2r\rho vN} \sum_{i=1}^{M} \frac{1}{i}
\end{aligned}
$$

for Random and Random-Repulse, respectively.

A. Simulation

We now resimulate our different techniques using our probabilistic sensor. The sensor probabilistically checks for targets every 128 ms. We choose our half-power distance to be 0.01 m; this gives the robots a probability of 53% of detecting an obstacle at a range of 10 cm and 32% of detecting an obstacle at 20 cm over the course of 1 second. We use a d_{max} of 20 cm.

B. Results

A bar chart of the average search time for the different techniques over 100 runs using different numbers of robots can be seen in Fig. 3. All techniques completed the search before timeout in all scenarios, but the average completion time is significantly higher than in the perfect sensing case. All three approaches achieve comparable results now. There are two likely reasons for this change. First, noisy detection increases the benefit of redundant coverage, as the target may not have been detected in the first pass, which nullifies much of the advantage of the Coordinated approach, since it may have to retrace its path multiple times. Second, because targets on the path periphery require a large number of expected passes with the Coordinated technique, there will occasionally be much higher completion times that increase the average.

Fig. 4. Expected completion time for 10 robots using probabilistic sensors with Coordinated and Random-Repulse algorithms
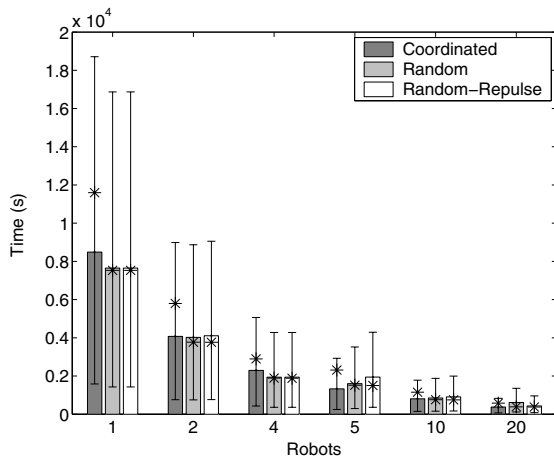

Fig. 3. Search performance of different techniques with noisy detection over 100 runs; asterixes represent analytical expected values and error bars represent 95% confidence intervals assuming an Exponential distribution

We can numerically calculate the expected completion times for the different algorithms. We find that $m \approx 2.8$ and $\rho \approx 0.33$. The adjusted expected completion times can be seem in Fig. 3 as asterixes overlaid upon the bars. There is close agreement between the Randomized simulations and analytical results. For the Coordinated approach, the simulated time was somewhat less than expected. This is likely because in the analysis, sensorial overlap between swaths was not taken into account, and therefore our expected estimate is higher than it should be. We can also compare the two strategies with varying values of d_h; Fig. 4 shows the expected completion time for 10 robots using probabilistic sensors with different half-power distances (all other parameters remain the same). The Coordinated approach does better for large half-power distances, where detection is similar to that of the ideal case, but its performance deteriorates more quickly than Random-Repulse, which does better for small d_h.

IV. UNCERTAINTY IN ROBOT POSITIONS

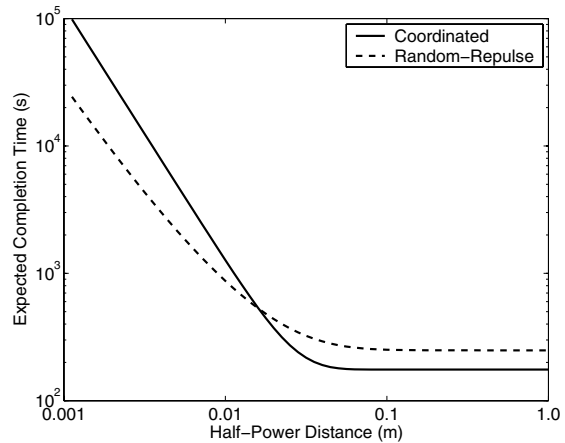Robots in the Coordinated technique require knowledge of their position within the arena in order to effectively navigate their prescribed path. Up until now, they have been given perfect positional knowledge. However, it can be very difficult to achieve very precise positioning depending on the environment. We consider several possible forms of position uncertainty.

Noisy Global Positioning: If robots are being informed of their position by some global system (e.g., GPS), there may be noise in the position they receive. This noise is often approximately Gaussian in nature and can vary over time. In order to simulate this effect, we add Gaussian noise to the perceived positions of the robots.

Cumulative Positional Noise: Depending on the scenarios, robots may not have access to a global system to provide them with positional information. In these cases, it may be necessary for robots to use their initial position knowledge and dead-reckoning with their odometry in order to maintain a position estimate over the course of the experiment. With dead reckoning, the uncertainty in position gradually increases over time as errors from wheel slip accumulate.

A. Simulation

The impact of positional noise on algorithmic performance is strongly affected by the geometry of the search environment (e.g., more confined environments would suffer more degraded performance than open ones). This increases the complexity of theoretical analysis, and we therefore only consider simulation of these effects here. We rerun our experiments (using perfect sensors) with positional noise.

Noisy Global Positioning: Every 4 seconds in the simulation, we generate a new error in the X position, Y position, and angle of each robot. These errors are sampled from a Gaussian distribution, with standard deviations of 40 cm, 40 cm, and 0.4 radians, respectively. This error is continuously applied to positional values until a new error is generated.

Cumulative Positional Noise: For each robot, the error in position is initially zero at the start of the simulation. At every time step of 128 ms, we add an adjustment to the error, sampled from a Gaussian distribution with standard deviations of 2 mm, 2 mm, and 0.002 radians for X position,

Y position, and angle, respectively. This gradually increases the error in perceived position on the robots.

B. Results

A bar chart of the average search time for the Co-ordinated techniques with noisy positioning compared to the Randomized techniques over 100 runs using different numbers of robots can be seen in Fig. 5. Not all runs were completed before timeout in this case with the cumulative noise Coordinated algorithm; 1 robot was only able to find the target 80% of the time, and 2 robots had a 96% success rate. In experiments with few robots, the Randomized approaches outperform the Coordinated approaches, especially with cumulative noise which takes very long to find the target on average. The poorer performance is due to positional noise causing robots wandering from their prescribed paths and therefore requiring multiple passes to detect the target. As the number of robots increases, the completion time increase factor remains approximately constant for noisy global positioning, but decreases significantly for cumulative positional noise, with 20 robots performing approximately as well as in the case with no positional noise. This is because cumulative positional noise has more drastic effects on robots which travel a long distance; if few robots participate in the search, the distribution of regions requires each robot to move farther, which increases the effect of the noise. In some cases, the robots believe their path leads to outside the arena bounds, which leads to deadlock and catastrophic failure, since they are never able to reach their next path point. The initial placement of robots in this scenario likely has a major impact on performance. If robots were required to align themselves from different initial positions, the performance would probably be much poorer, especially in the case with many robots.
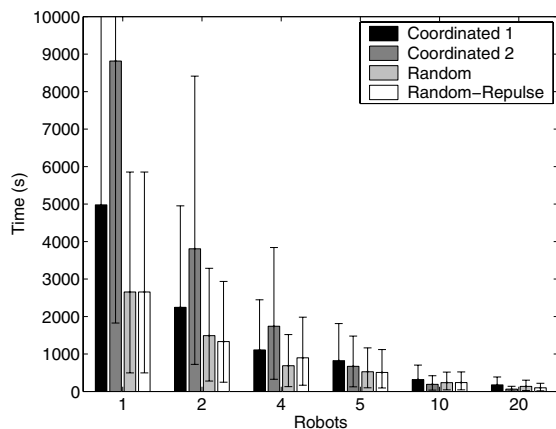

Fig. 5. Search performance of different techniques with noisy global positioning (Coordinated 1) and cumulative positional noise (Coordinated 2) over 100 runs; error bars represent 95% confidence intervals assuming an Exponential distribution

V. DISCUSSION

The analytical models presented here were able to predict with reasonable accuracy the performances of different algo-

rithms with different numbers of robots. These could be used to assist in the analysis and development of other multi-robot search scenarios, giving initial predictions of performance without needing to run full simulations. They could also serve as a basis for further analytical work on multi-robot search or other similar areas.

In many experiments that have been presented, the coordinated algorithm was still able to perform at least as well as the randomized one. This, however, does not necessarily mean that a coordinated algorithm is a better choice. By using a randomized algorithm, one opens the possibility of working with much simpler robots, which may not need to have as much sensorial capability or as much computing power. Design effort may also be saved, as it may take a substantial amount of time to produce a coordinated algorithm which works well for a given scenario. The benefit of faster performance must be weighed against these factors.

The coordinated algorithm used here is very simple, and could easily be improved in numerous ways. For example, robots with noisy detection could vary their paths on repeat coverage to better detect targets on the path periphery, and robots with cumulative positional noise could use a time-out/reset to prevent deadlock. However, even if safeguards are added to the algorithm to deal with potential pitfalls, it can be quite difficult or even impossible to predict every real-world problem that may arise. This is especially true for robots operating in dynamic or unknown environments (some algorithms may not even be able to function if the environment geometry is not known a priori).

One issue not dealt with here was the effect of robot failure. For randomized algorithms, other robots in the group simply continue their progress unhindered. For a coordinated approach, the group strategy must be recalculated to ensure that the area assigned to the broken robot is covered. This can be very difficult for the system and can even lead to catastrophic failure if the technique is not well-designed.

VI. CONCLUSION

Research on multi-robot systems often takes place using highly abstracted representations of the robots and their environment. While this can prove a valuable technique for speeding the design process, it is crucial to verify techniques using real experiments or simulations which incorporate real-world artifacts. We have shown that a high-performing coordinated approach to multi-robot search suffers degraded performance and sometimes even catastrophic failure when different real-world constraints are introduced. We have presented several randomized approaches which are much more robust to these affects.

While our randomized approaches showed superior robustness to the coordinated one presented here, this should not be taken as a blanket recommendation for purely randomized algorithms. Although the randomized algorithms used were able to eventually locate the target in all scenarios presented here, this may not occur in all cases. Some environments may be comprised of multiple convex regions connected by narrow passages. In cases like these, it would be necessary

to add some coordinated features to the algorithms in order to explicitly reach all areas of the environment. Our results should therefore be taken as a cautionary point on the design of highly-coordinated algorithms and as a demonstration of the need for realistic simulation/experimentation in multi-robot system design.

REFERENCES

[1] Acar, E. U., Choset, H., Yangang, Z., & Schervish, M. (2003) "Path Planning for Robotic Demining: Robust Sensor-based Coverage of Unstructured Environments and Probabilistic Methods", International Journal of Robotics Research, Vol. 22, No. 7-8, pp. 441-466.

[2] Benkoski, S. J., Monticino, M. G., Weisinger, J. R., 1991. "A Survey of the Search Theory Literature", Naval Research Logistics, Vol. 38, Nr. 4, pp. 469-494.

[3] Butler, Z. J., Rizzi, A. A., & Hollis, R.L. (2000) "Cooperative Coverage of Rectilinear Environments", Proc. of the IEEE International Conference on Robotics and Automation, San Francisco, CA, Apr 24-28, pp. 2722-2727.

[4] Cianci, C., Raemy, X., Pugh, J., & Martinoli, A. (2006) "Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics," Swarm-Robotics Workshop, Springer Lecture Notes in Computer Science. To appear.

[5] Cheng, C. K. & Leng, G. (2004) "Cooperative Search Algorithm for Distributed Autonomous Robots", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Sendai, Japan, pp. 394-399.

[6] Choset, H. (2001) "Coverage for Robotics - A Survey of Recent Results", Annals of Mathematics and Artificial Intelligence, Vol. 31, pp. 113-126.

[7] Gage, D. W., (1993) "Randomized search strategies with imperfect sensors", In Proc. of SPIE Mobile Robots, Boston, Vol. 2058, pp. 270-279.

[8] Gage, D. W. (1995) "Many-Robot MCM Search Systems", Proc. of the Autonomous Vehicles in Mine Countermeasures Symposium. Monterey, CA, pp. 4-7.

[9] Hayes, A. T., Martinoli, A. & Goodman, R. M. (2002) "Distributed Odor Source Localisation", Special Issue on Artificial Olfaction, Nagle H. T., Gardner J. W., and Persaud K., editors, IEEE Sensors Journal, Vol. 2, No. 3, pp. 260-271.

[10] Hayes, A. T. (2002) "How Many Robots? Group Size and Efficiency in Collective Search Tasks", Proc. of the 6th International Symposium on Distributed Autonomous Robotic Systems DARS'02, Fukuoka, Japan, pp. 289-298.

[11] Kantor, G., Singh, S., Peterson, R., Rus, D., Das, A., Kumar, V., Pereira, G., & Spletzer, J. (2003) "Distributed search and rescue with robot and sensor teams", Proc. of the 4th Intl. Conf. on Field and Service Robotics, Japan.

[12] Kurabayashi, D., Ota, J., Arai, T., & Yoshida, E. (1996) "Cooperative Sweeping by Multiple Mobile Robots", Proc. of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, Apr 22-28, pp. 1744-1749.

[13] Landis, G. A. (2003) "Robots and humans: Synergy in planetary exploration", Proc. of Conference on Human Space Exploration, Albuquerque, NM, Feb. 2-6, pp.

[14] Michel, O. (2004) "Webots: Professional Mobile Robot Simulator", Int. J. of Advanced Robotic Systems, Vol. 1, pp. 39-42.

[15] Parker, L. E. (2002) "Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets", Autonomous Robots, Vol. 12, No. 3, pp. 231-255.

[16] Polycarpou, M. M., Yang, Y., & Passino, K. M. (2001) "A Cooperative Search Framework for Distributed Agents", Proc. of the International Symposium on Intelligent Control, Mexico City, Mexico, pp. 1-6.

[17] Pugh, J. & Martinoli, A. (2006) "Relative Localization and Communication Module for Small-Scale Multi-Robot Systems", Proc. of the IEEE International Conference on Robotics and Automation, Miami, FL, May 15-19, pp. 188-193.

[18] Zheng, X., Jain, S., Koenig, S., & Kempe, D. (2005) "Multi-Robot Forest Coverage", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Edmonton, Canada, pp. 3852-3857.