Autonomous Cooperation of Heterogeneous Platforms for Sea-Based Search Tasks

Andrew J. Shafer[*], Michael R. Benjamin[†], John J. Leonard[*], Joseph Curcio[*]

[*] Massachusetts Institute of Technology
Cambridge, MA 02139
Email: ajshafer@alum.mit.edu
Email: jleonard@mit.edu
Email: jacurcio@alum.mit.edu

[†] NAVSEA Division Newport RI
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: mikerb@csail.mit.edu

***Abstract*— Many current methods of search using autonomous marine vehicles do not adapt to changes in mission objectives or the environment. A cellular-decomposition-based framework for cooperative, adaptive search is proposed that allows multiple search platforms to adapt to changes in both mission objectives and environmental parameters. Software modules for the autonomy framework MOOS-IvP are described that implement this framework. Simulated and experimental results show that it is feasible to combine both pre-planned and adaptive behaviors to effectively search a target area.**

I. Introduction

Previous research in the searching domain has focused on "lawnmower" style patterns of mapping an area [1]–[6]. However, this process deteriorates as more vehicles are added. If a vehicle is assigned to a portion of the overall area and then breaks down, the other vehicles must be manually reconfigured to scan that portion. Also, this process generally assumes the platforms are identical and does not take advantage of the capabilities that may be unique to each vehicle (e.g. one may have side-scan sonar, one forward-looking sonar, another optical or magnetic sensors, etc.). Finally, this process does not allow for cooperation with other objectives of the vehicle, such as collision avoidance or periodic surfacing.

Current search algorithms also generally assume a constant probability of detection for targets. This may not be realistic for real-world environments where the sea-state and sea-bottom can vary drastically. Previous research in AUV behavior development has primarily focused on implementing behaviors for a single vehicle [7]–[9], such as a single AUV taking temperature readings during ocean trials. With access to larger numbers of autonomous craft, research is needed to explore ways for vehicles to interact [10], [11], especially if vehicles have different capabilities (such as autonomous surface vehicles and autonomous underwater vehicles, [12]).

II. Background

A. A Brief Overview of Mine Countermeasures (MCM)

The ultimate goal of mine countermeasures (MCM) is to neutralize mines (called targets). There are two basic ways to go about this, minesweeping and mine hunting. Minesweeping refers to the process of neutralizing all mines in an area without specifically locating them first. This might involve, for example, using a helicopter to tow a large ferrous sled through mined waters, hoping to activate the mines so that they no longer pose a threat to area shipping traffic. Mine hunting, on the other hand, involves locating, detecting, and classifying mines for later neutralization. This neutralization might be done by the mine hunting vehicle itself, or it might be done by Navy divers or specialized vehicles. (For a more thorough introduction, see [13].)

The earliest iterations of mine hunting involved Navy divers locating underwater mines, marking their locations, and setting explosive charges for later detonation. This process is very accurate with regards to the detected targets but is extremely time consuming, expensive, and also dangerous for covering large areas. This process was improved with the invention of sonar which allowed ships to map the seafloor. Sonar data was then processed into a graphical form that a trained operator would look at to mark the mine-like objects. Mine hunting involves two steps; the first step is *detection*, where an oil drum, a large rock, and a mine should all be "detected". The second step is *classification*. Classification is determining which objects are mine-like and which are not. For this example, the oil drum and mine could both be considered mine-like. These two phases are defined as mine-hunting. The third phase of mine countermeasures is *identification*, which seperates mines (the targets) from non-mines (the clutter). The first three phases are sometimes referred to as DCI. The fourth phase, *neutralization*, eliminates the threat from the mines.

The maturing of autonomous underwater vehicles (AUVs), initially automated the data-collection phase of mine hunting by using AUVs programmed to follow a lawnmower style path over a search region, record data, and then return home. After retreival, data was offloaded where trained operators would look at the images to detect and classify mine-like objects. This automated data collection allows a large search area to be covered efficiently, but is still operator intensive. The next advancement in mine hunting detection and classification was using computer image processing to pre-screen the images, assisting the operator by pointing out obvious targets. The automated detection and classification of mine-like objects is called CAD/CAC (computer-aided detection/computer-aided classification). Recent advances in image processing, combined with space and energy efficient computers have now allowed CAD/CAC to be performed onboard the AUV. Currently, the only advantage that onboard CAD/CAC provides is that the AUV can communicate a "top-ten" list of targets *while still searching*, allowing vessels to begin identification and neutralization before the entire area has been scanned. The AUVs, however, do not use the information they have gained to alter their search patterns. An active area of research in this field is to try to close the loop and allow the AUV to autonomously make adjustments to its plan based on the information it is gathering.

B. Autonomous Underwater/Surface Vehicles (AUVs/ASV)

Autonomous underwater vehicles come in a variety of sizes, from 9-inch diameter man-portable units like Hydroid's REMUS [14] to Bluefin's 21-inch diameter BPAUV [15] (see Figure 1). Both the large and small units are typically equipped with acoustic modems, but the larger units are more likely to also have an inertial navigation system, more powerful sensors, larger batteries and possibly towed acoustic sensors. Both small and large AUVs are capable of carrying some form of sonar for bottom sensing, including mine-like object detection.

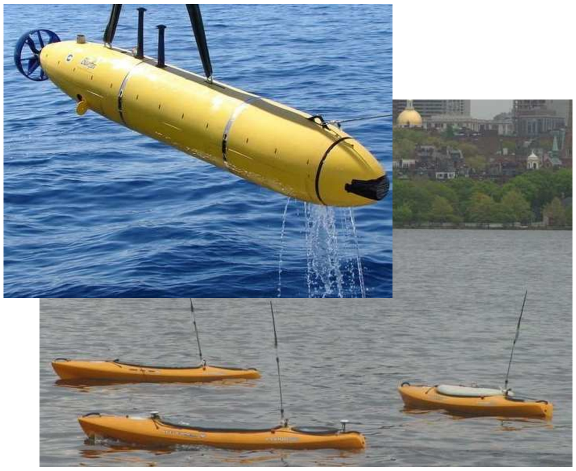


Fig. 1. Bluefin's 21-inch diameter AUV is capable of diving exceptionally deep for long periods of time to take readings near the ocean floor. Advanced autonomy software makes the independent operation of these vehicles possible. The SCOUT platform of autonomous kayaks are a relatively inexpensive robotic platform. The low cost and ease-of-use make it possible for research groups to own and frequently operate several of these vehicles.

Autonomous surface vehicles based on a kayak platform are also commercially available and inexpensive compared with typical AUVs. Four SCOUT platform [16] were used in this work, each equipped with with GPS, a compass, a thruster capable of nearly four knots, and an off-the-shelf PC for autonomy. Variants of the SCOUT have been fitted with such equipment as long-wave radios [11], acoustic modems [17] for communicating with underwater vehicles, and CTDs for taking water sound-velocity measurements. See Figure 1.

C. Autonomy Software

The autonomy software for this project is a suite of software referred to as "MOOS," the Mission Oriented Operating Suite [18], coupled with an additional suite of autonomy software and autonomous helm known as the IvP Helm [19]. The collective suite is referred to as MOOS-IvP. Fundamentally MOOS is middleware that allows running processes to communicate in a publish-subscribe manner. This community of software modules typically runs on the payload computer of a marine vehicle, sending and receiving information from the main vehicle computer. This arrangment is known as the backseat driver paradigm, and is illustrated in Figure 2.

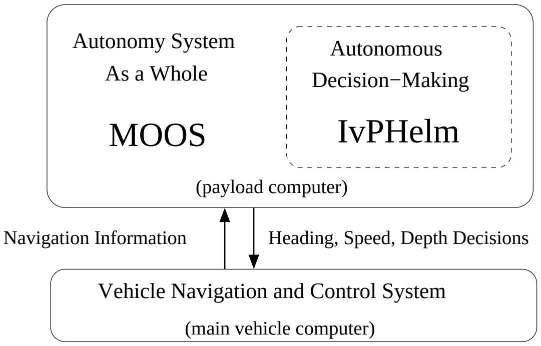


Fig. 2. The backseat driver paradigm separates vehicle autonomy from vehicle control. The autonomy system provides heading, speed and depth commands to the vehicle control system. The vehicle control system executes the control and passes navigation information, e.g., position, heading and speed, to the autonomy system.

An example community of software modules used in this work is shown in Figure 6. The IvP Helm is a single MOOS process known as pHelmIvP. It uses a behavior-based architecture to implement autonomous decision making. Each behavior produces an objective function over the vehicle decision space (*heading*, *speed*, and *depth*), and the helm performs multi-objective optimization to reconcile behavior output. Example behavior objective functions are rendered in Figure 7.

Both MOOS and IvP are independent but coordinated Open Source software projects maintained by Oxford University, MIT, and the Naval Undersea Warfare Center. Several MOOS modules and IvP Helm behaviors used in this work are available as part of the public software distribution (www.moosivp.org). Two new MOOS modules and two new IvP Helm behaviors, described in a later section, were designed and implemented to address the goals of the problem described in this work.

D. A Novel Approach to Adaptive Autonomous Search

Current algorithms for mine hunting assume two features of the search vessel, (1) Mine hunting is the only task the vehicle is trying to complete, allowing the search algorithm to completely determine the actions of the vehicle. (2) The sensor's detection ability is constant for all regions of the search area and does not depend on environmental conditions or vehicle state (these parameters are referred to as the sensor's A and B values, described later). These two assumptions preclude using an adaptive algorithm for mine hunting. By violating these assumptions a novel solution is proposed for decomposing the search area into smaller cells that allows for adaptive autonomous mine hunting. In particular, the algorithm is adaptive to changes in (a) the mission, by allowing multiple objectives to be completed simultaneously, and (b) the environment, by adjusting the sensor's A and B parameters.

III. Mine Countermeasure Theory

Planning for mine countermeasures (MCM) operations takes on a probability-based framework that trades the risk of missing targets for the time spent to cover the search area [20], [21]. The current objective of this planning operation is to determine the number of search tracks, N, across the search area of channel width C to attain the desired clearance level P (average probability of neutralizing mines) (see Figure 3).

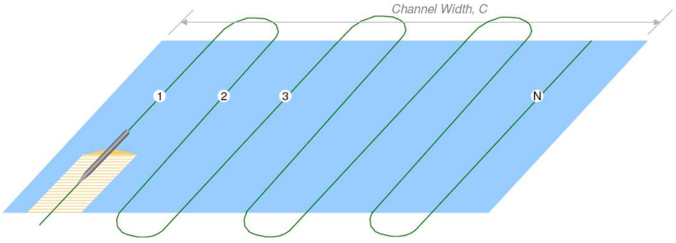


Fig. 3. When planning for a mine countermeasures operation, the mission commander specifies a desired clearance level P, for the search area. The result of the planning is N, the number of search tracks to cover the entire channel width, C. Image courtesy Rafael Rodriguez.

The probability that at any instant a given searcher detects a given target depends on the relative locations of the searcher and target, the physical capabilities of the sensor, the physical characteristics of the target, and the environment. Because the probability of detection is very low for long distances between the searcher and target, it is possible to calculate the probability of detection for a target at a given lateral range (distance between a target the straight line trajectory of a searcher) by integrating the instantaneous probability of detection from $-\infty$ to ∞. See Figure 4.

For our purposes then, a sensor can be completely described by its nominal detection probability, B, and its nominal range (width), A, as depicted in Figure 4(d).

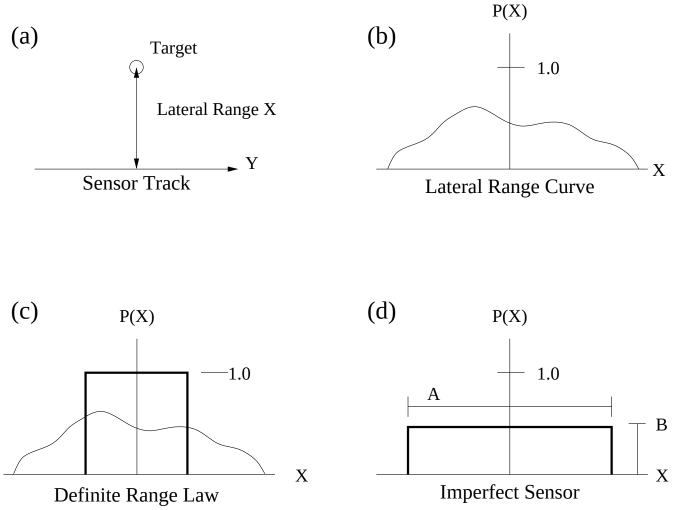


Fig. 4. In (a), the sensor passes along a target at lateral range X. Integrating the instantaneous detection probability (a function of X and sensor and target characteristics), we obtain a fictional (b), the lateral range curve. This curve goes to 0 for large X and may not be symmetric, depending on the sensor configuration. Approximations for (b) include (c), which assumes that all targets within a fixed distance are detected (cookie cutter), and (d), which more accurately models the sensor's performance (imperfect sensor). In (d) the scalar A represents the sensor range and B the probability between $[0, 1]$. Image (slightly modified here) originally appears in [20].

The mission commander for an MCM operation specifies a desired clearance level, P, which gives the probability that any particular mine has not been neutralized. For example, if the commander desires 96% clearance, that allows a 4% chance each mine has not been neutralized. In a field of 100 mines, it would be acceptable for four of them not to be neutralized.

With a perfect sensor ($B = 1$), the clearance level is just equal to the portion of the search area covered by the sensor (see Equation 1). For an imperfect sensor, the probability of missing any particular mine is inversely proportional to the number of times the entire area has been scanned.

$$P = \begin{cases} \max(N * A/C, 1) & B = 1 \\ 1 - (1 - B)^{N*A/C} & B < 1 \end{cases} \tag{1}$$

Further complicating clearance operations is that the probability of identification $P_{id} < 1$, the probability of neutralization $P_n < 1$, and there are some mines that cannot be detected, μ. If the commander desires an overall clearance level of P, we can correct for these factors by increasing the coverage to

$$P' = P/(P_{id} * P_n * (1 - \mu)) \tag{2}$$

Because we can correct for these factors by increasing P, we will ignore any complications caused by these factors. In the case where $P > (P_{id}*P_n*(1-\mu))$, no number of paths can satisfy the desired clearance level and a later minesweeping effort will have to be conducted.

A. Applications to Small Cells

The previous discussion applies to a large search area where it makes sense to talk about the number of paths that cover the search area. It is possible to apply this reasoning to small cells, which is at the heart of what we are trying to do.

Recall that the clearance level, P, is an average of $1-$(the probability that an undetected target is present). Also recall that B is the probability that the sensor will detect a given target as it passes by the target. It is important to note the assumtion that detection events are independent for each target and for each pass of the sensor over the target. However, this assumption might not hold for some targets or environments. For example, some targets have a strong directionality which makes them easily detectable from one direction but not from another. Applying this to a small cell, every time a cell comes into view and then leaves, we must update the clearance level of that cell. Assuming detection independence, the equation for updating a cell that has been sensed is in Equation 3.

$$P_{new} \quad = \quad 1-(1-P_{old})(1-B), \tag{3}$$

where $(1-B)$ is the probability a target is missed on this pass, $(1-P)$ is the probability it was missed on a previous pass, and P_{new} is new updated probability that a target has been missed. Figure 5 shows a graphical procession of what happens when a sensor is passed over a cell multiple times.

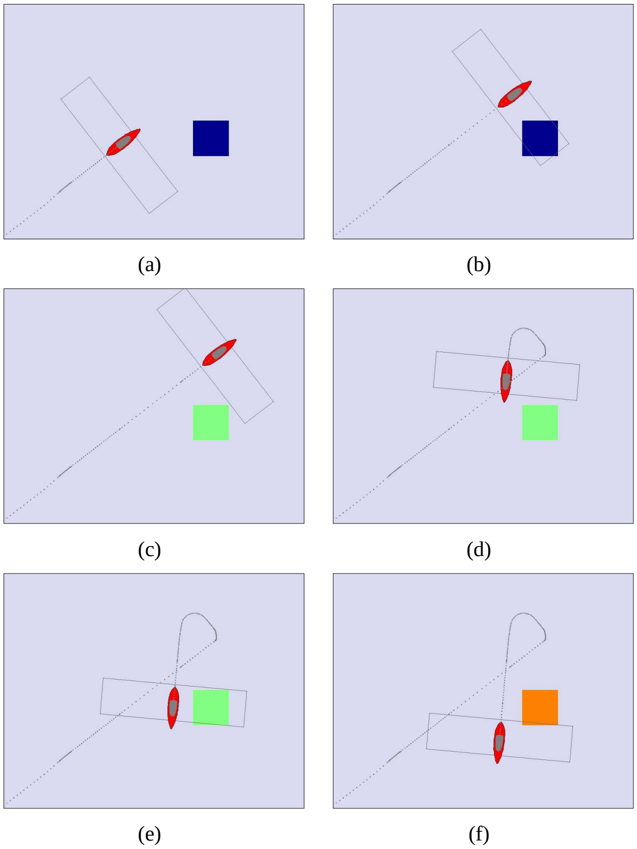


Fig. 5. In this figure, a sensor is passed over a single cell. As the sensor enters the cell (b), nothing is changed. When the sensor has completely passed the center of the cell (c), its clearance is updated according to Equation 3. Here, the cell has changed from 0 clearance (blue), to .5 clearance (teal). On the second pass, (d)-(f), the cell is updated from .5 to .75 (orange); which is two passes of a .5 sensor. These updates are communicated to other searchers to adjust their own internal grids.

The average clearance for a group of cells is the average of each cell's clearance weighted by its relative area. For equal-sized cells, this is equivalent to summing the clearances and dividing by the number of cells.

$$\text{Average Clearance } P = \frac{\sum_{i=1}^{N} P_i}{N}$$

B. Signal Detection Scoring Metrics

The scoring metric used to evaluate a potential search algorithm greatly influences the design of that algorithm. For instance, with a metric that greatly punishes missing targets and only lightly penalizes incorrectly declaring a target, the algorithm may decide to declare many of the objects it sees as targets. In addition to the standard signal detection theory scoring (see Table I), the score might be affected by factors such as the time of search (penalizing long search times), the clearance level, the number of search vessels used, amount of energy expended, or the completion of certain objectives, such as finding a lane [22] from one side of the area to the other.

While not explored in this research, it would be beneficial to create a scoring metric that uses classic signal detection theory but that also incorporates each cell's clearance level, P, when assigning points. For example, if a cell has a high clearance level, meaning that not many targets should be missed there, the penalty for misses would be increased. To simplify the calculation of scores, we use classic signal detection scoring. We will hold the other possible scoring variables (such as time or clearance level) constant to compare search algorithms.

TABLE I

BASIC SIGNAL DETECTION THEORY TERMINOLOGY.

	Target Present	Target Not-Present
Declare Target	Hit	False Positive
Declare No Target	Miss	Correct Rejection

IV. SOFTWARE FRAMEWORK AND MODULES

The principle development work for this project is the creation of three software modules: the *pSensorSim* and *pArtifactMapper* MOOS modules and the *SearchArtifact* helm behavior. These software modules form a chain that starts with simulated targets and ends with desired heading and speed commands for the vehicle. See Figure 6 for a diagram of the relationship. The pSensorSim module loads the artifacts to simulate and publishes the artifacts that it detects. The pArtifactMapper subscribes to these artifact updates and forms a clearance map of the area of interest. The search behavior then subscribes to this clearance map and rates decisions about where the vehicle should go next. Any of the modules can be replaced with components that follow the same interface. For example, the sensor module could be replaced with the actual sensor hardware, and the search behavior could be replaced with any other search algorithm.

A. The pSensorSim MOOS Module

The pSensorSim module is a MOOS process that simulates output of a fictional sensor given a "threat laydown," which

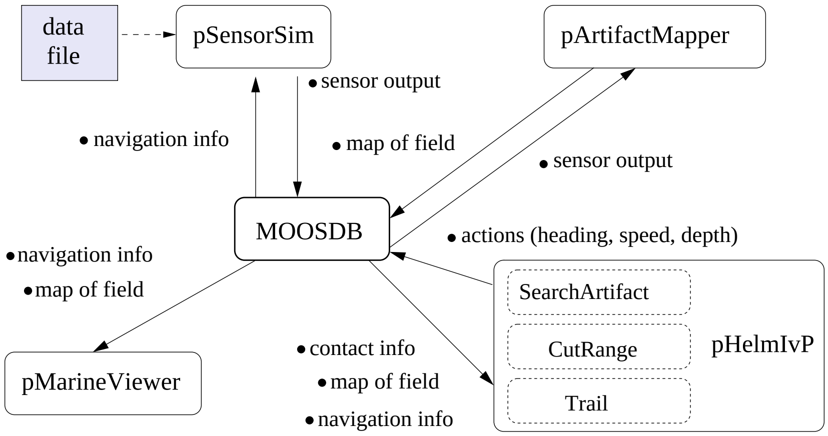


Fig. 6. This diagram illustrates the flow of data between the two MOOS modules *pSensorSim*, *pArtifactMapper*, and the *SearchArtifact* behavior contained in the pHelmIvP MOOS module. The pSensorSim models a simple sensor and outputs the artifacts that it detects. The pArtifactMapper module remembers this output and also keeps track of which cells the vehicle has covered, and produces an artifact map by publishing periodic updates. This artifact map is used by the SearchArtifact behavior to produce an objective function for the helm to consider in its action selection process. Other standard helm behaviors such as the CutRange and Trail behavior may also be active in the helm.

is a list of targets read in from a file. The basic process is to maintain a list of artifacts that can currently be "seen" by the sensor. On each iteration, for each *new* artifact, pick a random number [0, 1). If less than B (the sensor detection capability), post the artifact as detected for processing by other modules. Now store the list of all seen artifacts for the next iteration. Another function of pSensorSim is to simulate variations in the sensor's A and B values. Recall that A is the effective width of the sensor and B is the probability that the sensor will detect a given target. pSensorSim uses a simple two-value sensor B parameter that changes the sensor's normal value to a reduced value for a short period of time. For example it might have a B value of $.8$ for 60 seconds and then drop to $.25$ for 10 seconds before repeating the cycle. There is further work in modeling a sensor's B value given the vehicle state and bottom environment.

B. Search Behaviors

The search algorithm is arguably the most important component of an adaptive search platform. For example, certain sensors, such as side-scan sonar, require the vehicle to travel in a fairly straight line in order to get good data. Other sensors, such as magnetic field gradiometers, have little dependency on turn rate. A search algorithm that tries to turn frequently would result in poor data from the side-scan sonar but would not affect the magnetic field gradiometer. Conveniently, in this framework the search algorithm is also the most replaceable component. Because of the prevalance of side-scan sonar for mine hunting tasks, we will assume that turns are to be avoided to minimize lost data collection opportunities.

The work presented here utilizes two vehicles, one that mimics an AUV and performs a simple lawnmower behavior, and a second vehicle that mimics an autonomous surface craft (ASC). This scenario is plausible because when an ASC and AUV opperate in cooperation, both vehicles benefit. The AUV gets accurate location updates from the ASC without wasting time and energy to surface, and the surface craft is able to act as a relay for communications with the AUV.

In this work, these vehicles cooperate by having the AUV perform a standard lawnmower pattern over the search area while the ASC tries to perform two objectives. First, it needs to stay close to the AUV for reliable communications and location updates. Second, because it knows which cells that still have low clearance levels, the ASC can try to go out of its way to cover those cells. The IvP Helm interval programming method for reconciling behavior output allows us to find a solution for these competing objectives. See Figure 7.

1) The SearchArtifact Behavior: The SearchArtifact behavior uese a simple greedy algorithm that chooses the desired heading and speed based on which choice gives the most delta in coverage. The delta for an individual cell is:

$$\Delta P \quad = \quad B(1 - P) \tag{4}$$

Intuitively, this makes sense, because ΔP is just the effect of passing a B-grade sensor over an area that has a $(1 - P)$ probability of having something undetected in that cell. Then for any given speed and heading, we can calculate which cells will be covered within a time horizon, and add the ΔP values for each cell in that set to get the utility of travelling in that heading at that speed.

C. The pArtifactMapper MOOS Module

The role of the pArtifactMapper MOOS module is to maintain two data structures: 1) a list of all the detected artifacts, and 2) a clearance level associated with each cell of the search grid.

The *search area* is the physical region of interest, defined by a convex polygon. The search area polygon is completely tiled with equal-sized, square elements, forming a *search grid*. The size of the cells is specified by the user. In this work 5-meter squares are used, to be small enough to be within the error

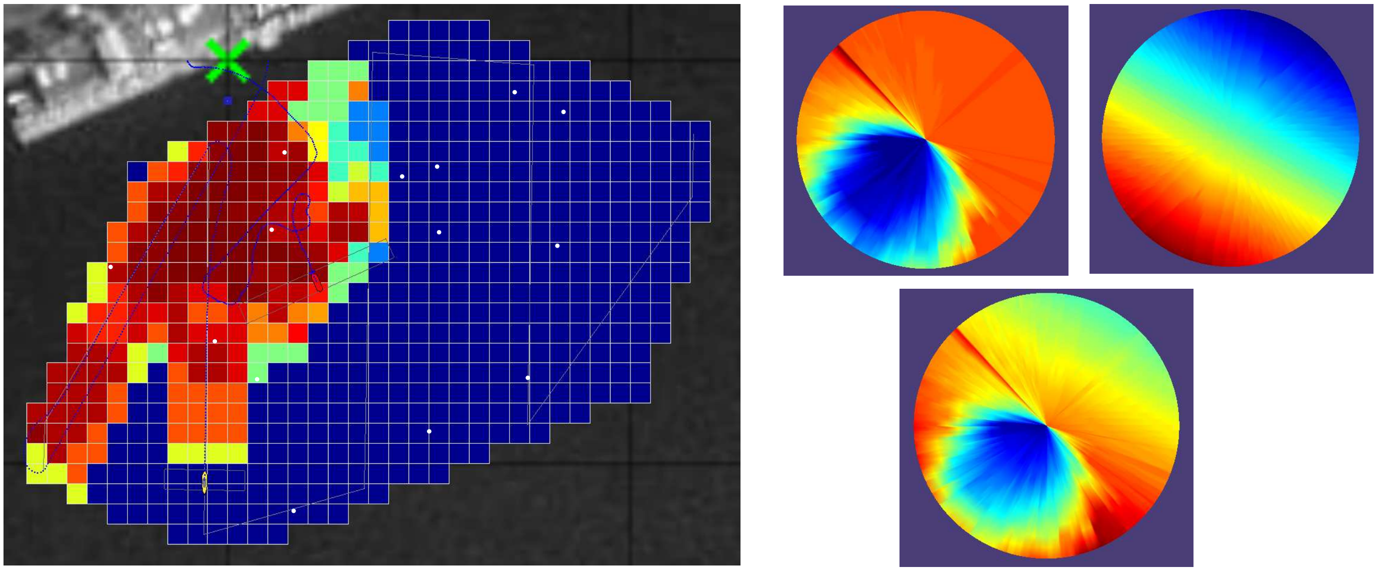


Fig. 7. This example illustrates how multiple objectives can be satisfied simultaneously by the interval programming method of pHelmIvP. The operation area shown is the MIT Sailing Pavilion (the structure in the upper left). The white dots are artifacts known to pSensorSim from the threat laydown file. The color of each cell represents its clearance level, dark blue is 0 clearance, dark red is nearly 100% clearance. The dark blue trails behind the vehicles represents the path that vehicle has travelled. The yellow vehicle (near the bottom) is running a lawnmower behavior. The red vehicle (near the middle) is running both a search behavior and a cut range behavior to the other vehicle. The objective functions in play for the red vehicle are shown on the right. The two upper plots represent objective functions produced by the ArtifactSearch and CutRange behaviors respectively. The plots are defined over heading and speed with higher speeds further out radially from the center. Colors represent the utility with red representing high utility and blue lower. The ArtifactSearch behavior is expressing desired headings and speeds that cover the unexplored cells. The CutRange behavior is expressing desired headings and speeds that take the vehicle closer to the target vehicle. The influence of this behavior (weight) increases with increasing distance, making it more important when the vehicle is far from the target. The radial plot on the bottom is the weighted sum of the other two functions. The helm efficiently finds the peak of this collective function. In this case, the desired behavior is to travel at top speed (1 meter/sec) at heading 156.

bounds of a sensor (approximately 8-m [23]), but large enough that the number of cells is manageble. Other constraints might limit the maximum number of cells that can be described. For instance, if an AUV allocates 10-bits to describe the cell index number, the number of cells is limited to 1024 (2^{10}), regardless of an optimal size for the cell. Also, square cells might not be the best shape for the elements. Some discretization methods use hexagonal cells because of their close-packing ability and because it removes ambiguity as to which cells are considered neighbors [4]. When pArtifactMapper receives a detected artifact notice from pSensorSim, it stores the artifact in the cell corresponding to the artifact's X and Y values.

Using the theory developed in Section III-A, for cells that have entered and left the sensor's effective area (i.e., those cells have been scanned), those cells have their clearances updated.

$$P_{new} \quad = \quad P_{old} + B(1 - P_{old})$$

After updating a cell, this information needs to be communicated to other searchers as a grid update. Grid updates take the form of index, prev_clearance, new_clearance. The previous clearance is included in the update in case platforms want to try to reconstruct a sequence of grid updates. To communicate these updates to other vehicles, other MOOS processes deliver the message in whatever form is appropriate. For communication over TCP/IP channels, pMOOSBridge forms a connection to both MOOSDBs and pushes updates in one direction. Underwater, acoustic modems are used to communicate data at slow rates.

1) The Lawnmower Behavior: The Lawnmower behavior is a component of the IvP helm that implements a lawnmower pattern generation algorithm. It is basically a waypoint following behavior that takes as inputs the parameters of the pattern, search area, path radius, and path heading.

V. Simulated and Experimental Results

In our experiments two kayaks of the SCOUT platform were used [16]. Each was equipped with Garmin 5 Hz GPS sensors for location information. For each mission (summarized in Table III), the scoring metrics in Table II were used. Missed targets were heavily penalized (reflecting a high cost of not knowing about targets) while false alarms are mildly penalized (reflecting the relative ease of re-evaluating targets). Because our sensor does not produce actual false alarms, the overall score for an algorithm is directly proportional to the average clearance level. A constant threat-laydown of 30 uniform-randomly placed targets in a polygon of 15,100 m^2 was used.

TABLE II

Scoring metrics used in experimental evaluation

HitPoint:	10
MissPoint:	-1000
FalseAlarmPoint:	-10
CorrRejPoint:	10

TABLE III

SUMMARY OF SIMULATIONS AND EXPERIMENTS. TIME IS IN MINUTES:SECONDS

#	Scenario Description	Time	Score
1	A fast-moving search kayak follows a slow-moving AUV. Both vehicles have a 20-m, .5 B sensor with drops to .25 B for 10 seconds out of 30. We discover that slew-rate limiting does not function properly during experiments because of the noisy heading data.	16:42 21:21	-1090 (sim) -1820 (exp)
2	Same as first scenari but we turn off the slew-rate limit. This allows us to collect data, but experiments overstate the ability of the sensor by over-scanning cells because of noisy heading data.	16:11 22:00	1940 (sim) 3960 (exp)
3	We remove adaptivity from the vehicles by having both perform lawnmower patterns over the search grid. This holds time constant to allow us to compare clearance levels.	17:11 24:00	-2060 (sim) 3980 (exp)
4	We again remove adaptivity from the vehicles but this time each vehicle runs a pattern over half of the search area.	11:00 14:00 12:00	-2100 (sim) -2100 (exp1) -6120 (exp2)
5	The lead AUV performs a lawnmower pattern with a 40-m wide, .5 B sensor while the kayak follows behind with a 20-m wide .8 B sensor.	10:48 12:52 12:45	1980 (sim) 3010 (exp1) 930 (exp2)
6	We reverse the previous experiment and have the kayak lead with a wide sensor while the slower AUV follows with the more accurate sensor. Experiment 1 used a 10-m swath for the lawnmower pattern while Experiment 2 used a 20-m swath.	14:09 16:52 7:37 10:07	970 (sim) 5980 (exp1) -2100 (sim2) -40 (exp2)

In all, seven mission simulations and nine experimental missions were analyzed. Missions were divided into six scenarios shown in Table III for both simulated and experimental testing of different parameters for cooperative, autonomous searching. In the time column, faster is better. Simulations were typically faster because the vehicles have a more difficult time turning on the water. In the score column, higher is better. Because our scoring metric gave -1000 points for each missed target, missing a few targets quickly put some scores into negative values. The first four scenarios all used the same vehicle and sensor parameters. In the missions of the fifth and sixth scenarios, one sensor was made to be wider and the other sensor was made more accurate. Detailed analysis for each mission type is presented in the sections.

A. Scenario 1: An AUV with a Kayak Tender

In this scenario we model an AUV being followed by an ASC tender (in this case, an autonomous kayak). As described earlier this scenario is plausible because it allows the kayak to provide reliable, fast communications between mission control and the AUV. Both vehicles use the A-B sensor model described earlier with a $A = 20$ meters and $B = 0.5$ probability dropping to $B = 0.25$ for 10 seconds every 30 seconds. The slew-rate limit feature of pSensorSim and pArtifactMapper is set at 10 degrees/sec. The sensor and mapper return no results when the vehicle is turning beyond this limit. Slew-rate is defined as the difference in heading data points divided by the time difference between those two points. For noisy data collected at a fairly fast rate (5 Hz GPS sensor), the noise quickly overwhelms the slew-rate. Vehicle 1, the AUV, travels at 1.0 meters/sec and is setup to perform a lawnmower pattern with a 10-m radius (half the sensor width). Vehicle 2, the kayak, travels at 1.5 meters/sec and runs both SearchArtifact and CutRange behaviors. Table IV contains the results for Scenario 1.

TABLE IV

SIMULATION AND EXPERIMENTAL RESULTS FOR SCENARIO 1.

	Simulation	Experiment
Score:	-1090	-1820
Time:	16:42	21:20
Average Clearance:	77.9%	18.7%
Actual Clearance:	70.0%	13.0%
Hits/Misses:	21/7	4/24
False Alarms:	4	1
Correct Rejections:	574	577

In this scenario, the sensor and artifact mapper are configured with a slew-rate limiter that prevents them from outputting data when the vehicle is turning faster than 10 degrees/sec. In simulation, this works fine. In the experiment on the water, the noise in the heading data causes no data to be returned. The heading noise was due to faulty compasses on the day of testing and reliance instead on the GPS for heading information, which is relatively inaccurate at low speeds. Note how in simulation the clearance was a reasonable 78%, but that in the actual water mission the performance was a miserable 18.7%. This result shows the importance of trying to experimentally validate results that have been shown in simulation. For the rest of the missions we were forced to turn off the slew-rate limit. This also means that the rest of the missions over-state the clearance of cells that pop in and out of the sensor range because of heading noise.

B. Scenario 2: An AUV and Kayak Tender, No Slew-Rate Limit

In this scenario we repeat the parameters from Scenario 1 but with the slew-rate limit turned off. The vehicle now scans cells more easily, especially while turning or when the noisy heading reading causes cells to “pop” in and out of view of the sensor. This cell “popping” occurs because when the slew-rate is multiplied by the half-width of the sensor, the distance that the tip of the sensor moves can be significant. This moving edge can quickly cover and uncover a cell, causing pArtifactMapper to update the clearance of that cell multiple times. This effect could be corrected by cleaning the heading data or using some other metric to determine when a cell has been cleared. Figure 8 shows the effect of turning slew-rate limiting off for both the simulated mission and the experiment. Table V summarizes the results.

TABLE V
SIMULATION AND EXPERIMENTAL RESULTS FOR SCENARIO 2

	Simulation	Experiment
Score:	1940	3960
Time:	16:11	22:00
Average Clearance:	76.7%	87.1%
Actual Clearance:	80.0%	86.6%
Hits/Misses:	24/4	26/2
False Alarms:	4	4
Correct Rejections:	574	574

Comparing the results in Tables IV and V, we see that the effect of turning off the slew-rate limit is minimal for simulations (77.9% vs. 76.7%) but significant for experiments (18.7% vs. 87.1%).

C. Scenario 3: Two Vehicles with Full Lawnmower Patterns

In this mission the vehicles do not adapt to the search grid. Each vehicle runs a lawnmower pattern, one with paths at 90 degrees, one with paths at 0 degrees. This allows us to hold time constant to see how the clearance level differs from adaptive missions. The results are summarized in Table VI.

TABLE VI
SIMULATION AND EXPERIMENTAL RESULTS FOR SCENARIO 3

	Simulation	Experiment
Score:	-2060	3980
Time:	17:11	24:00
Average Clearance:	70.7%	84.6%
Actual Clearance:	66.7%	86.6%
Hits/Misses:	20/8	26/2
False Alarms:	2	3
Correct Rejections:	576	575

The results in Tables V and VI are compared to assess gains in using an adaptive behavior over just regular lawnmower. In simulation, the adaptive mission produced a clearance of 76.7% in a time of 16:11. The pure lawnmower mission produced a clearance of 70.7% in about 17:11. Similarly, on the water the adaptive behavior cleared 87.1% in approximately 22:00 while the pure lawnmower cleared 84.6% in approximately 24:00. In both cases an incremental gain in both time and average clearance is seen.

D. Scenario 4: Two Vehicles Partitioning the Search Area

In the previous mission time was held constant by having both vehicles cover the entire search grid. In this mission we reduced the time of search by splitting the search grid in approximately half and having one vehicle lawnmower over each portion. When each vehicle finished its half, it returned to the dock. Each vehicle had used a simulated A-B sensor with $A = 20$ meters (sensor width) and $B = 0.5$, dropping to $B = 0.25$ for 10 seconds every 30. The standard CollisionAvoidance behavior of the IvP Helm was also activated. The results are summarized in Table VII.

TABLE VII
SIMULATION AND EXPERIMENTAL RESULTS FOR SCENARIO 4

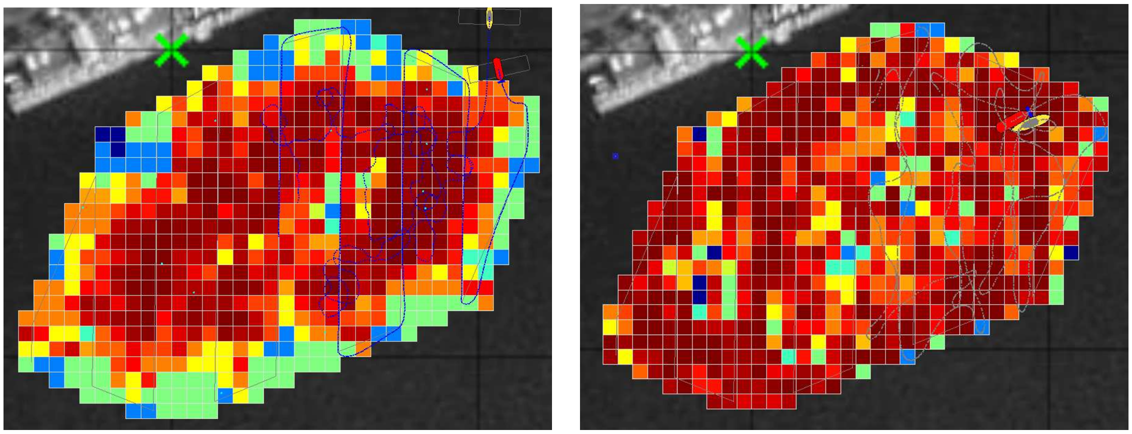
	Simulation	Experiment 1	Experiment 2
Score:	-2100	-2100	-6120
Time:	11:00	14:00	12:00
Average Clearance:	50.0%	61.5%	57.6%
Actual Clearance:	66.7%	66.7%	53.3%
Hits/Misses:	20/8	20/8	16/12
False Alarms:	4	4	3
Correct Rejections:	574	574	575

Experiment 1 and 2 are two different runs of the exact same mission profile. The cause of the lower score for run 2 is unknown. The average clearance is reasonably close to that of run 1, but the actual clearance is much lower. Because of the small target set (30 mines) and the heavy penalty for missing targets (-1000 points), small deviations from the mean are heavily represented in the score. Ideally, each cell in the grid would be covered once by a pass of a sensor. This would give us an expected clearance of $.5 * 20/30 + .25 * 10/30 = .41667$. The 50% coverage in simulation is probably caused by overlap from the two vehicles near the center of the search grid and from when the vehicles turn at the end points. The even higher coverage in the experiments is caused by the noisy heading data over-scanning cells near the periphery of the sensor.

E. Scenario 5: AUV Leading with a Wide Sensor

Part of the adaptability aspect of this work is to provide for cooperation when the vehicles have different sensors. In this scenario both vehicles use a simulated A-B sensor. The AUV has an $A = 40$ meter wide swath, with $B = 0.5$ again with 10/30 second drops to 0.25. The kayak has $A = 20$ meter wide swath, with $B = 0.8$ with 10/30 seconds dropt to 0.6. The slower AUV performs a lawnmower patter, and the faster kayak performs the CutRange and ArtifactSearch behaviors. The results are summarized in Table VIII.

Experiment 1 was conducted with standard IvP Helm CollisionAvoidance active on both vehicles. Experiment 2 turned off the lead vehicle's collision avoidance so that it would not be disturbed by the chase vehicle. The difference between experiments 1 and 2 are minor for the average clearance and time (the difference in actual clearance is likely a statistical anomaly). This means that adding collision avoidance to the vehicles did not impact mission performance but in real applications could prevent the loss of vehicles from collisions.



(a)

(b)

Fig. 8. In this figure, the sensor and artifact mapper are configured without the slew-rate limiter. (a) shows the simulation results while (b) shows the experimental results.

TABLE VIII

SIMULATION AND EXPERIMENTAL RESULTS FOR SCENARIO 5

	Simulation	Experiment 1	Experiment 2
Score:	1980	3010	930
Time:	10:48	12:52	12:45
Average Clearance:	81.0%	82.7%	85.8%
Actual Clearance:	80.0%	83.3%	76.7%
Hits/Misses:	24/4	25/3	23/5
False Alarms:	2	1	4
Correct Rejections:	576	577	574

F. *Scenario 6: Kayak Leading with a Wide Sensor*

In this scenario the kayak has the wide sensor and is the lead vehicle that performs a lawnmower pattern. The AUV has a narrower, more accurate sensor. This conceivable configuration could result from a single mine hunting surface craft that might be responsible for one or several AUVs. The AUVs would have to both stay near the surface ship for tending while simultaneously exploring interesting/less-covered areas. The results for this mission are in Table IX.

TABLE IX

SIMULATION AND EXPERIMENTAL RESULTS FOR SCENARIO 6

	Sim # 1	Exp #1	Sim # 2	Exp #2
Score:	970	5980	-2100	-40
Time:	14:09	16:52	7:37	10:07
Average Clearance:	85.9%	95.8%	69.3%	78.6%
Actual Clearance:	76.7%	100.0%	66.7%	76.7%
Hits/Misses:	23/5	28/0	20/8	22/6
False Alarms:	2	4	4	2
Correct Rejections:	576	574	574	576

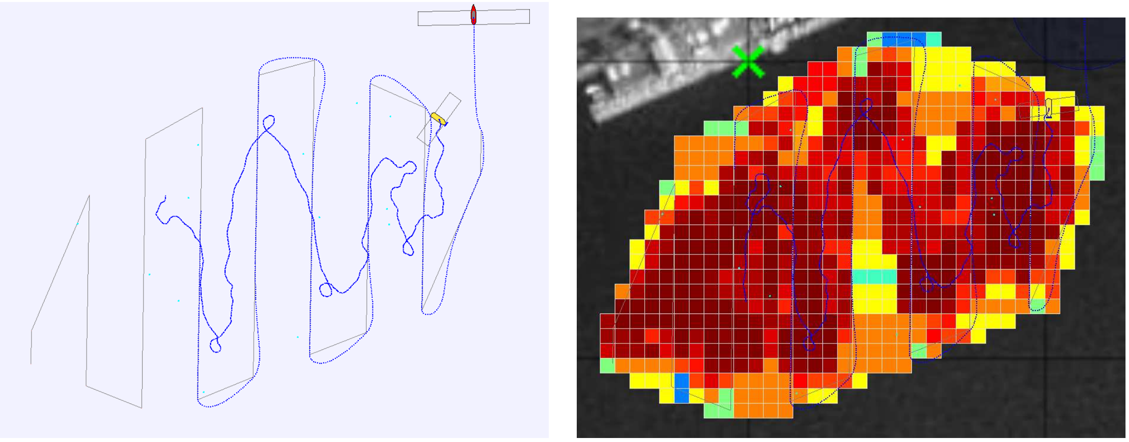
In Experiment 1, a lawnmower radius of only 10 meters was used, for a path-to-path distance of 20 meters, half of the lead sensor width. Thus the lead sensor covered every part of the search area twice, which is why the clearance level is so high and the time is 70% longer than Experiment 2. Experiment 2 used a radius of 20 meters. Comparing the two scenarios, the faster lead vehicle achieves the anticipated decrease in search time but also a decrease in coverage. In simulation we increase the coverage by 11% but increase the time by 41%. Experimentally we see a 4% increase in coverage (compared to the previous experiment 1) for a 27% increase in time. Figure 9 shows that the AUV spent much of its time trying to catch up to the lead vehicle. That is, the cut range behavior had significant influence on the vehicle.

VI. CONCLUSIONS

The goal of this work was to investigate the feasibility of using autonomous marine vehicles to cooperatively search a given area. To that effect we have shown that effective cooperative search is possible, even when it can be difficult to evaluate the effectiveness. Evaluation of cooperative search algorithms is difficult because of the number of simultaneous objectives. Time of search, energy used, number of vehicles, number of targets detected, lane-finding, and others are all reasonable ways of scoring a search algorithm.

The greedy search behavior described here seems to function well when it cooperates with another behavior that guides it towards exploring the whole search area, such as following a lawnmower-based lead vehicle. This algorithm assumes targets have a uniform distribution and the utility in clearing any one cell is the same as any other cell. This means the average clearance (based on the cells) is usually close to the actual clearance (based on the total targets found). This assumption is often used in MCM planning since it simplifies calculations used for analytic solutions. A more intelligent behavior could be written that exploits patterns that naturally exist in deployed mine fields, such as lines or clusters of mines.

Our sensor model also assumes that the B value of the sensor is uniform along its fixed width, A. We assumed



(a)

(b)

Fig. 9. When the lead vehicle can travel faster than the following vehicle, the follower spends much of its time playing catch-up, (a). In this image the actual vehicle paths are in dark blue. The waypoints for the lead vehicle are represented by the grey line. The final clearance map for simulation 1 is pictured in (b). Notice that the gap of least coverage in the bottom-center of the image occurred where the chasing vehicle was trying to catch up to the lawnmowing vehicle. In the clearance map, dark red is heavily cleared areas while dark blue, light blue, teal, yellow, and orange represent increasing levels of clearance.

constant B along the sensor width because we had no model for another distribution. By modelling random sensor dropouts we can begin to understand how a non-uniform B might affect the search, but more research into sensors is needed.

An issue with CAD/CAC is the prevalence of false positives. Our simulated sensor does not emit false positives but could by adding fake targets to the threat laydown. We did not evaluate the effect of false positives. Further experiments should focus on testing how different parameters affect the search performance. Some suggested parameters include the relative weights of SearchArtifact and CutRange behaviors, the distances at which the cut range behavior becomes active, and the sizes and shapes of the search areas.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the Office of Naval Research for this work, in particular the support of the IvP Helm basic research provided by ONR Code 311, Dr. Don Wagner and Dr. Behzad Kamgar-Parsi under N0001408AF00002.

REFERENCES

[1] E. Acar, H. Choset, Y. Zhang, and M. Schervish, “Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods,” *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 441–466, 2003.

[2] H. Choset, “Coverage for robtics–a survey of recent results,” *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 113–126, 2001.

[3] S. Hert, S. Tiwari, and V. Lumelsky, “A terrain-covering algorithm for an AUV,” *Autonomous Robots*, vol. 3, no. 2, pp. 91–119, 1996.

[4] E. Stoneking and J. Hosler, “Path planning algorithms for the adaptive sensor fleet,” *Advances in the Astronautical Sciences*, vol. 121, pp. 207–217, 2005.

[5] I. Wagner, M. Lindenbaum, and A. Bruckstein, “Smell as a computational resource-a lesson we can learn from the ant,” *Proc. ISTCS*, vol. 96, pp. 219–230, 1996.

[6] ——, “Distributed covering by ant-robots using evaporating traces,” *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 5, pp. 918–933, 1999.

[7] M. Benjamin, J. Curcio, J. Leonard, and P. Newman, “Navigation of unmanned marine vehicles in accordance with the rules of the road,” *International Conference on Robotics and Automation (ICRA)*, 2006.

[8] M. Benjamin, M. Grund, and P. Newman, “Multi-objective optimization of sensor quality with efficient marine vehicle task execution,” *International Conference on Robotics and Automation (ICRA)*, May 2006.

[9] M. Benjamin, J. Leonard, J. Curcio, and P. Newman, “A method for protocol-based collision avoidance between autonomous marine surface craft,” *Journal of Field Robotics*, vol. 23, no. 5, pp. 333–346, 2006.

[10] T. Balch and R. Arkin, “Communication in reactive multiagent robotic systems,” *Autonomous Robots*, vol. 1, no. 1, pp. 27–52, 1994.

[11] J. Rice, “Undersea networked acoustic communication and navigation for autonomous mine-countermeasure systems,” *Proceedings of the 5th International Symposium on Technology and the Mine Problem*, 2002.

[12] R. Williams, “Design and experimental evaluation of an autonomous surface craft to support AUV operations,” Master’s thesis, MIT, 2007.

[13] G. Hartmann *et al.*, *Weapons that Wait: Mine Warfare in the US Navy*. Naval Institute Press, 1979.

[14] *The Hydroid website. [online].*, 2008, available: http://www.hydroidinc.com/pdfs/remus100web.pdf.

[15] *The Bluefin website. [online].*, 2008, available: http://www.bluefinrobotics.com/bluefin_21bpauv.htm.

[16] J. Curcio, J. Leonard, and A. Patrikalakis, “SCOUT–a low cost autonomous surface craft for research in cooperative autonomy,” *IEEE Oceans*, 2005.

[17] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, “The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms,” *IEEE Oceans Conference*, 2005.

[18] P. Newman, “MOOS - a mission oriented operating suite,” Massachusetts Institute of Technology, Department of Ocean Engineering, Tech. Rep. OE2003-07, 2003.

[19] M. Benjamin, “A guide to the IvP helm for autonomous marine vehicle control,” Massachusetts Institute of Technology, Tech. Rep., 2006.

[20] D. Gage, “Randomized search strategies with imperfect sensors,” *Proceedings of SPIE Mobile Robots VIII*, vol. 2058, pp. 270–279, 1993.

[21] R. Rodriguez, “Unmanned systems autonomy and mcm performance,” Technical Talk, March 2008.

[22] S. Redfield, “Lane finding using homogeneous groups of cooperating autonomous vehicles,” *Autonomous Underwater Vehicles, 2004 IEEE/OES*, pp. 26–31, 2004.

[23] R. Stokey, T. Austin, B. Allen, N. Forrester, E. Gifford, R. Goldsborough, G. Packard, M. Purcell, and C. von Alt, “Very shallow water mine countermeasures using the REMUS AUV: A practical approach yielding accurate results,” *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, vol. 1, 2001.