Sensor Abstractions to Support Many-Robot Systems

Douglas W. Gage

RDT&E Division, Naval Command Control and Ocean Surveillance Center
NRaD Code 531, San Diego, CA 92152-5000

ABSTRACT

The rapid evolution of micromechanical fabrication techniques and other sensor, effector, and processing technologies will soon make it possible to employ large numbers of very inexpensive autonomous mobile robots with fairly limited sensor capabilities to perform real-world missions in the ground, air, space, and underwater environments.

One approach to such a system is to realize desired emergent collective group behaviors with simple sensor-based reactive planners. The initial thrust of this effort has been to develop generic ensemble behaviors, such as blanket, barrier, and sweep coverage, and various deployment and recovery modes, which can address a broad spectrum of generic applications, both military and civilian. However, while different applications may require similar group behaviors, the sensor, information, and communications resources available to the participating individual robots may be very different.

This paper outlines the many-robot approach to real-world problem solving and discusses the various roles that different types of sensors can play in such systems. Analysis and simulation results are presented to show how useful behavioral algorithms can be designed to make use of diverse information resources, and the area search problem is analyzed to derive both system measures of effectiveness and system design considerations.

1. INTRODUCTION

Rapidly evolving sensor, effector and processing technologies, including micromechanical fabrication techniques, will soon make possible the development of very inexpensive autonomous mobile devices with adequate processing but fairly limited sensor capabilities [1, 2]. The general notion of "swarms" of "insect robots" has received attention in the popular and semipopular press [2, 3, 4], and "microrobots" have even been joined by "nanorobots" in the popular imagination [5]. The use of large numbers of small robots (with varying degrees of "large" and "small") has been seriously proposed for a wide variety of applications, including intelligent land mine deployment [6], behind-the-lines military communications relaying [7], warehouse security sentry [1], ship hull cleaning [1], warehouse material handling [4], lunar base construction [8], gathering oceanographic data [9], planetary surface exploration [10, 11, 12], and aircraft carrier deck foreign object debris (FOD) disposal [13].

Nature, of course, provides outstanding models of functioning systems consisting of large numbers of more or less intelligent and mobile elements. One example is found in the flocking, herding, and schooling behaviors observed in many different types of vertebrates. While the literature on mammal herds, bird flocks, and fish schools is mostly descriptive or coarsely analytic [e.g., 14, 15], a synthetic approach to the study of such group behaviors was pursued by Craig Reynolds [16] in order to develop a realistic-looking animation sequence of a flock of birds. More interesting than vertebrate flocking, however, are the behaviors of the social insects: ants, bees, and termites; the observed aggregate behaviors exhibit a greater complexity, while the individual animals are much simpler. Through experimental manipulation of insect colonies and computer simulations, researchers have elucidated some of the mechanisms by which these colonies survive and grow by adapting to their changing environment. For example, Deneubourg [17] has demonstrated via simulation that sorting behaviors observed in ants can be produced by the simplest possible biasing of random behavior by environmental cues, while Franks [18] has used simulation to explore the changing raiding patterns of army ants. Seeley [19] has investigated how worker honey bees appropriately initiate various productive activities in response to quite simple signals and cues. Honey bee colonies thus provide a model for achieving "purposeful" coordinated group action, responsive to changing environmental conditions, without employing a world model -- in fact, without explicit global decision making of any sort.

With various individual and group animal behaviors serving as "existence proofs", quasi-intelligent "emergent behavior" resulting from the interaction of simple reactive planners has been proposed as the basis for the intelligent control of individual robots, in the development of usefully complex systems [20] as well as simple conceptual vehicles [21]. The term "Swarm Intelligence" has been used to describe the application of this approach to distributed systems consisting of perhaps hundreds of elements [22]. The limitation of nature's "existence proofs", however, is that the "purpose" of a natural system is to survive and reproduce, so that specific behaviors that appear purposeful to the observer merely represent larger or smaller hills in the topographical fitness map continuously processed by the forces of natural selection. The realization of emergent behaviors to allow an artificial system to achieve an *a priori* specified purpose may not always prove to be a straightforward matter.

A key prerequisite to realizing real world mission goals through the collective behavior of large numbers of relatively simple, inexpensive, interchangeable, autonomous elements is the capability to control the system of robots in terms of meaningful mission-oriented system-level parameters. A user (the military term is "commander") requires an understanding of a system's capabilities, doctrine for employing it, and measures of effectiveness to assess its performance once deployed. It is thus necessary to relate system (ensemble) functionality and performance to the behaviors realized by the individual autonomous elements [13].

2. THE COVERAGE PARADIGM

Many potential applications for many-robot systems involve the performance of some function which can be characterized as "coverage": the application of the effects of some sensor or effector to some extended physical space. Potential applications of coverage behaviors include mine deployment, mine sweeping, reconaissance, sentry duty, communications relay, maintenance inspection, carrier deck FOD disposal, and ship hull cleaning. In each case it is necessary to develop precise measures of effectiveness which meaningfully characterize the overall system performance in the context of specific mission goals. For example, a surveillance group should be large and sparse if the goal is to maximize the number of targets detected per unit time over a wide area, but small and dense to minimize the probability of leaving any targets undetected within a smaller swept area. Three varieties of useful coverage behaviors can be distinguished:

Blanket (or Field) coverage: The objective is to achieve a static arrangement of elements that maximizes the detection rate of targets appearing within the coverage area.

Barrier coverage: the objective is to achieve a static arrangement of elements that minimizes the probability of undetected penetration through the barrier.

Sweep coverage: the objective is to move a number of elements across a coverage area in a manner which addresses a specified balance between maximizing the number of detections per time and minimizing the number of missed detections per area. (A coordinated sweep can be roughly equivalent to a moving barrier, but sweep coverage can also be achieved using random uncoordinated element movements; see the analysis of section 5.)

To achieve optimal coverage, the desired ensemble behavior is the maintenance of a spatial relationship between system elements which is appropriate to the mission sensors or effectors, and which adapts to specific local conditions. The basic approach is to do with robots what is actually done today with humans in applications such as aircraft carrier deck foreign object debris (FOD) disposal: "You guys line up at arm's length distance, and then all walk along together, picking up whatever you find as you go". The motion of each element is based on the motions of the other elements, more strongly on the motions of the nearest neighbors, but with some reference to the more remote elements as well. Thus, navigational coordination is achieved by using sensor inputs (each element sensing the position of its neighbors, relative to itself), vice explicit communications channels.

In addition to coverage behaviors, which represent the "bulk steady state" behaviors of the system, it is necessary to consider and to provide for various relevant spatial and temporal "boundary condition" behaviors, such as deployment, recovery, and navigation of the group as a whole. Other factors to be considered include obstacles and traversability, randomness of behavior, and the applicability of analogies from biology (herding, schooling, immune system and pheromone mechanisms) and physics (entropy, temperature, pressure, solid, liquid, gas) in analyzing the internal dynamical states of the system [13].

The defining feature of the systems considered in this work is that the number of mobile robotic elements is large enough that the system command/control interface must "hide" the individual elements from the user/commander of the system. The elements don't have to be small or inexpensive, but economic and technical factors make it highly likely that they will be. The level of abstraction is such that collisions between elements or of individual elements with discrete obstacles are not dealt with: a collision avoidance mechanism may function transparently, or, alternatively, the elements may be considered to be small enough that collisions between them are very unlikely, robust enough that collisions don't damage them, or expendable enough that damage to or loss of elements doesn't matter at the system/mission level.

Our abstract system, then, consists of a large number of identical elements, although this stricture can, of course, be relaxed to permit multiple *castes* of elements, analogous to those of the social insects. Each element possesses: (a) some measure of mobility -- this may be fairly limited, so that, for example, in an underwater application the elements may be capable of regulating only their depth in the water column whle drifting with the currents; (b) some mission-capable sensor or effector, which actually performs the desired mission function; (c) optionally, some navigational sensor capability that allows each element to measure, at least crudely, its position with respect to at least its nearest neighboring elements and/or in some reference coordinate system (this sensor may be the same as the mission sensor capability listed above); (d) optionally, some communications capability, which may make use of the sensor capability; and (e) some processing capability, which implements algorithms to navigate the element so that that the ensemble of mission-capable sensors or effectors collectively accomplish the desired mission objectives (this may involve maintaining a specified positional relationship to its neighbors).

3. SENSOR ABSTRACTIONS FOR NAVIGATION

Just as nature has provided interesting models for many-element systems, she has also provided a rich set of models for sensor capabilities and, at least as instructive for our purpose, models for how to use different types of sensor inputs to achieve desired navigational behaviors. However, because natural systems are the product of natural selection and not of purposeful design, we must be careful in applying the models of natural systems to the artificial systems we design to achieve our *a priori* chosen goals. Viewed from our goal-oriented perspectives, in some biological systems nature appears to be imaginative beyond belief, while in others she seems incomprehensibly blind to obvious opportunities. In this section we consider various aspects of the role of sensors in the basic problem of navigation: moving an element from where it is to where it is supposed to be.

Let's consider the precise meaning of that phrase "where it is supposed to be": the terms of reference in which the creature/mission "goals" are framed. A biological system is focused on satisfying some goal referenced to aspects of its (their) internal state(s) -- a *self-referenced* goal (e.g., hunger). Some sub-goals may be referenced to some features of the creature's immediate environment -- an *environmentally-referenced* goal (e.g., hungry, therefore looking for food). On the other hand, an artificial system can be assigned a mission which may be referenced to abstract aspects of a more "global" reality -- an *externally-referenced* goal. A mission step may consist simply of "go to position LATLONG and wait for the next instruction." This is just the analogy in the mobility domain of the classical robot manipulator controller, which generates a synchronous stream of commands prescribing the next actuator position for each manipulator degree of freedom. Clearly, the control paradigms we are interested in will have more kinship with the biological model than with the robot manipulator controller.

The most fundamental sensor capability required for an element to move to "where it is supposed to be" is the ability for a randomly wandering element to recognize (with some probability) that place when it gets there, but even this is an absolute requirement only if the element has to recognize when it has completed this "mission step" in order to perform other functions (e.g., eat food once it has found it).

The next level of sensor capability is one that not only detects when it is where it is supposed to be, but can also, within some parameters, tell whether the element is getting "warmer" or "colder" as it moves. This could correlate with nearer or farther in terms of physical distance from the target position, or to any other gradient that can be followed to the target. A shark can follow a scent gradient to find a bleeding prey; a single photocell can be used to efficiently guide a robot to a lit beacon.

A sensor with the next level of sophistication provides an indication of the absolute direction to the target (e.g., by processing the inputs from an array of photocells, the robot can immediately point itself right at the lit beacon). If the vehicle navigation controller is to use the sensor in a mode more sophisticated than the "warmer - colder" paradigm, then a *data representation* is needed to communicate the preprocessed sensor output to the controller so that the controller can establish the relationship between "sensor space" and "actuator space" necessary for its internal control loop.

In an actual system implementation, the difference between these last two control modalities might exist only in the location of the boundary between the "sensor preprocessing" and "vehicle navigation processing" boxes on a high-level system block diagram. The notion of data representation becomes much more important, however, when the absolute direction sensor is being used to provide a *reference* direction, instead of a *target* direction. While many biological systems make extensive use of sensors to provide target directions (*tropisms*, based on light, gravity, magnetic field, etc.), far fewer use reference directions. A notable example is the worker honey bee's navigation using the polarization direction of the sky as a reference, and the associated use of the angle of the honey bee "dance" (referenced to the vertical) to communicate the direction to a food source (referenced to the sky polarization).

Since physical considerations of element mobility capability (including energy consumption and environmental traversability) often determine the appropriate speed of an element, the lowest level navigation function often consists of sending the desired direction of motion ("course") to the lowest level vehicle controller (engine and steering). Thus, it is directional sensors which are primary at this level of navigation, while range information is principally used at a higher, computational level, except, of course, while maneuvering close to other objects such as obstacles or rendezvous partners. (Given a straight course and a known speed, it is of course possible to calculate the range to a stationary object from the rate of change of its bearing, and its bearing -- with ambiguities -- from the rate of change of its range.)

If we look again to nature's example, however, we see that biological systems generally "don't do arithmetic", and, in general, minimize the communication of representations of quantitative sensor data ("readings"), instead using sensor data to effect tight closed loop control. Sensor "preprocessing" is "hardwired", with multiple "preprocessors" often operating on the same raw data type (as in the foveal and peripheral vision systems). Regulation to a setpoint is often achieved by balancing off competing mechanisms. Nature is often extremely clever at "computing through the environment" to create cues for appropriate behavior, as cited in [17-19]. *Pheromones* provide an important example of animals arranging their environment to, for example, help moths find mates, or to help a colony of ants effectively exploit a food source that one ant discovers.

The pheromone concept is one which might be applied to many-robot systems; for example, tiny retroreflective glass beads might be used like ant trail pheromone. Pheromones represent just one example of using sensors in a *cooperative mode*; *beacons* are another. Cooperative sensors are vulnerable to subversion in two ways, however; the insect world provides examples of predators homing in on other species' mating pheromone, and also of predators emitting "forged" mating pheromone (or firefly beacon pattern) to lure dinner to them. Thus, the use of cooperative sensors raises important security issues, especially in military systems [23, 24]. An ideal pheromone would use an encrypted interrogation technique.

Let us now return to the possible sensor roles identified in the abstract model of a many-robot system presented at the end of section 2: mission performance, neighbor element coordination, (global) navigation, and communications. In a coverage application, the mission sensor provides a natural scale factor for the desired interelement spacing by operating the sensor in a cooperative mode so that elements which are too close are distinguishable from those which are not -- the processing becomes simplest when the sensor's detection threshold is the discriminator. Even if the mission involves effector coverage, rather than sensor coverage, it is likely that the effector will have a sensor associated with it that might serve this purpose. (Of course, we must bear in mind the security caveats presented previously.) If elements are too spread out from one another, then directional (bearing) sensors can be used with algorithms such as that simulated in section 4 to help them "condense".

4. SIMULATION PROGRAM AND INITIAL RESULTS

A many-robot simulation program has been written on the Macintosh using Symantec's Think C compiler and associated development environment, while minimizing the use of Macintosh-specific constructs in order to maximize portability of the program to platforms with greater computational power. The simulation initially models a 2-dimensional world, and each element knows the exact relative positions of each of its fellows; neighborhood models and sensor capabilities and limitations (radially symmetric at first) will be implemented in the second iteration of the simulation. Ultimately, the model will be further generalized by incorporating a third dimension and building more detailed models of sensors and sensor pre-processing.

The program supports the creation of up to ten independently controllable behavior groups of elements (robots), with up to 200 elements total. The program user creates clusters of elements by specifying how many elements are to be created in each cluster, in what pattern (random distribution within a circle or rectangle of specified size and location), with what initial velocities (random distribution within ranges of speed and course), and with what behavior (zero acceleration, specified speed and course, rendezvous at specified time and position, or one of several sensor-based flocking algorithms). The simulation navigates each individual element in terms of speed and course; realizing the requested speed and course in terms of throttle and steering is allocated to an unsimulated lower level controller, which also navigates around point obstacles, such as other elements. The behaviors of the different groups of elements can be changed by user command as the simulation progresses. A simulation run can be saved as a file containing a succession of kinematic frames, and replayed later.

Figure 1 presents a sequence of several frames from a simulation run, showing a group of 40 elements executing a "condensation" behavior, coming together from an initially dispersed configuration. The actual algorithm is very simple: if the minimum azimuthal angle subtended by all an element's neighbors is less than 180 degrees (i.e., if it is possible to draw a line through the element so that all its neighbors fall on the same side of the line), then the element knows that it is "on the edge" of the group. Elements "on the edge" move at fixed speed down the bisector of the subtended angle; elements not "on the edge" remain stationary. Thus, the configuration "implodes", with the "outer" elements "sweeping" the remaining nodes insward as the condensation continues. The angle-bisecting component of this algorithm has been previously described by Sugihara and Suzuki [25], whose paper discusses a number of other interesting motion coordination algorithms. In a real application, a complementary (or in some implementations, "competing") behavior would halt the condensation process as the desired element density is achieved.

The condensation algorithm and its simulation present some interesting features for discussion. First, the algorithm is highly robust, in the sense that, even if the sensor range is much smaller than the initial diameter of the configuration, the algorithm will produce a single compact group, as long as the sensor detections produce a connected graph. (If the graph consists of disjoint pieces, each piece will condense to form a separate group.) This is true because any element in motion is moving closer to the positions of all other elements that it can see. The second point is that the condensation process requires no global position information of any sort; the command to the group is not "condense to position LATLONG", it is just "condense". The third point is that the algorithm makes use of only azimuthal information from sensors, and has no requirement for range data. However, algorithms generally similar in effect can also be designed using range data only, or a mix of range and azimuth data, and this is an area of exploration for the continuing simulation effort. The fourth point is that, in its initial instantiation, with a discontinuity in element motion based on a binary decision of "on the edge" or "not on the edge", the time-step simulation introduces an artifact of clustering at the "corners" of the configuration, as the different elements in a "corner cluster" play leapfrog, taking turns being "on the edge". It is possible, however, that the concentration of forces introduced by this "artifact" might be considered a desirable feature in some combat situations, and the algorithm could be time-quantized to produce it with real (vice simulated) robots.

5. SEARCH STRATEGIES: MEASURES OF EFFECTIVENESS AND SYSTEM DESIGN CONSIDERATIONS

The designer of a many-robot system will face the opportunities and challenges of working within a design space providing many degrees of freedom. Required system level functionality and performance may be achievable with many different system configurations, and the designer must make appropriate choices concerning the system as a whole (e.g., the number of different castes of elements, the populations of each, the desired ensemble behaviors, and the command and control organization of the ensemble), and the capabilities of each element type (e.g., effectiveness and range of mission sensors and effectors, vehicle platform limitations such as speed, endurance, weight, power source, and communications capabilities and programmed behaviors). In this section we explore some of the system design dimensions and the resulting system tradeoffs in the context of a single generic coverage application: area search.

Minesweeping is the archetypal area search application: a two dimensional area of interest (of area A) is suspected to contain some number of objects of interest (targets), and the search task is to detect these targets. For the purposes of this analysis, the targets are assumed to be stationary. While the basic problem addressed (in a number of interesting variations) by the field of optimal search theory [26, 27] is to devise strategies to optimally allocate a given set of search resources, our purpose in the following simple analysis is to develop a framework for choosing an optimal set of robotic resources from within a very rich design space, given specified operational goals and cost constraints.

The system we deploy consists of some number N of identical robotic elements, each of which can move about while carrying a sensor of nominal range r and detection probability p: any target which lies within a distance r of the element's track is detected with probability p as the element passes by. Targets farther than r are never detected, false alarm detections are disregarded, and what action the element might take with detected targets is of no concern. Each element is capable of traveling a total distance d during the mission; this limitation may be due to limited energy storage, or to operational constraints on the duration of the mission, coupled with the maximum speed which can be achieved while operating the sensor payload. The task is to detect at least a prescribed fraction D of the targets.

The assumption that p is strictly less than 1 is critical as the basis of the analysis that follows. In fact, the interesting case is when D > p, so that we need (on average) to sense each point more than once. This is consistent with our underlying assumption that our robotic system elements are simple and inexpensive. And, in any event, while an expensive sensor might *promise* a p very close to 1, we could never be *certain* that a single sweep of the search area would find all the targets. Let S be the number of times on average that each point in the search area is sensed; we call S the "sweep fraction" of the search, and calculate it as

$$S = N * 2r * d / A$$

As an initial case, let us assume that the elements are capable of executing a perfectly coordinated search pattern. We may have only one element following a "lawn-mower" pattern, perhaps by employing a highly accurate navigational system, or we may have a number of elements moving in a coordinated formation. Then we can calculate

$$D = 1 - (1 - p)^{S} \qquad (1)$$

In fact, this equation is true only for integer S, with straight line interpolation between integers, but we won't worry about this approximation for now -- the approximation gets very good as S increases. It is important to note that, for a given p and S, no other search strategy can perform better than this fully coordinated strategy. We can see this clearly in the case where we employ a single element: at each point in time, the element is sampling within the region where undetected targets are most likely to be found, namely those points which have been sampled the fewest times already.

As a second case, we could employ less expensive elements, capable only of staying within the designated search area, but otherwise wandering completely randomly. In this case, we find that some points are sensed a greater number of times, and some not at all, but we can still make use of the sweep fraction S as defined above. Now we find that

$$D = 1 - e^{-p * S} \qquad (2)$$

Unlike equation (1), equation (2) is accurate for non-integer S. Figure 2 shows the behaviors of equations (1) and (2) when p = 0.8.

We now consider the question of how much "better" is the completely coordinated search (first case) than the completely random search (second case)? We make the notion of "better" a precise one by recasting the question as: for a given sensor effectiveness p, how much larger a sweep fraction Sr is required so that the detect fraction D of a completely random search (second case) is equal to that of a completely coordinated search (first case) with sweep fraction Sc? Equating (1) and (2) above, we find that

$$Sr/Sc = -\ln (1 - p) / p \qquad (3)$$

This result says that, for any sensor detection probability p < 1, we can achieve any desired detect fraction D equally well by (a) performing a completely coordinated search with sweep fraction Sc we can calculate from equation (1), or by (b) performing a random search with sweep fraction Sr calculated from equation (2), which is larger than Sc by a factor (equation (3)) which depends only on p, and is independent of the desired detect fraction D and the S required to achieve it. Figure 3 shows how this ratio varies on p; it is fairly small for moderately imperfect sensors, and, as is obvious from inspection of equation (3), the numerator of the expression dominates the behavior as p approaches 1.

The fact that Sr/Sc is independent of the desired D suggests that we might be able to make use of the corresponding ratio to quantitatively describe the relative effectiveness of other search strategies. Accordingly, we define the "search gain" G of any given search strategy s as:

$$Gs = Sr / Ss$$

In other words, the search gain Gs of a search strategy s is the factor by which that strategy reduces the search fraction required to achieve any desired detect fraction, compared to a random search. So we can then write the detect fraction Ds for any search strategy s as:

$$Ds = 1 - e^{-p * Gs * Ss} \quad = 1 - e^{-p * Gs * N * 2r * d / A} \qquad (4)$$

G will be most meaningful if it depends only on p, as in the case of the fully coordinated search of case 1, but it will still be useful if it varies slowly and predictably with S. Note that while the G calculated for the completely coordinated search -- equation (3) -- serves as the maximum value achievable by any search strategy, it is entirely possible for a strategy's G to be less than 1.0 if element paths are positively correlated, as in the case of ants following each others' pheromone trails. A necessary next step will be the determination of the G function for other simple search strategies, such as having two elements which meet alter their courses to leave at right angles to each other (which would seem, intuitively, to negatively correlate their paths).

The reason for choosing to write equation (4) in this format is that the variables in the expression essentially provide a coordinate system for the system design space, and they break naturally into three groups:

> p and r are characteristics of the primary mission sensor: its single-pass probability of target detection and its range. Note that in some cases the choice of a sensor will have immediate implications for other system characteristics; for example, p and/or r may depend critically on element speed, as with sonar self noise.
>
> N and d, which together with r determine the sweep factor S, are parameters pertaining to the element platforms: the number of elements to be employed, and the effective search range of each. Mission time and stealth requirements, maximum platform speed, and energy storage limitations may be important in determining d.
>
> G is the gain in search effectiveness due to the coordination (*negative* correlation) of element search paths to provide balanced coverage; this is where vehicle search strategy behaviors are accounted for.

Figure 4 presents a (fictitious) simple application example in which the design space consists of 12 possible systems, allowing any of three possible sensors, a completely random or a completely coordinated search strategy, and an optional battery upgrade to double vehicle range, with each choice having an associated specified cost. As the quality of the sensor (its raw target detection probability p) improves, the most cost effective design shifts from employing a large number of the least expensive and least capable elements to a much smaller number of more expensive and more capable elements. In the real world, with a much more complex design space and unavailable, unreliable, and expensive cost estimates, the design process would probably begin with the determination of p for the mission sensor package, since determining p determines the maximum G that can be achieved. The lower the value of p, the more likely it is that it will be more cost effective to utilize a random search strategy (G = 1) and increase S, rather than implement a coordinated search strategy, with its added cost and complexity. Once the search strategy is selected, determining G, then the tradeoff between sensor range, number of elements, and search range per element can be made to realize the required S in the most cost effective fashion.

sensor			G*S	coord	coord?	double	cost /	num	cost of	
cost	range	detect p	reqd	gain G		range?	elem	elems	system	
1	1	0.5	9.21	1.39			11	460.52	5066	<---
1	1	0.5	9.21	1.39		yes	26	230.26	5987	
1	1	0.5	9.21	1.39	yes		41	332.19	13620	
1	1	0.5	9.21	1.39	yes	yes	56	166.10	9301	
10	1	0.7	6.58	1.72			20	328.94	6579	
10	1	0.7	6.58	1.72		yes	35	164.47	5756	<---
10	1	0.7	6.58	1.72	yes		50	191.25	9562	
10	1	0.7	6.58	1.72	yes	yes	65	95.62	6216	
15	1	0.9	5.12	2.56			25	255.84	6396	
15	1	0.9	5.12	2.56		yes	40	127.92	5117	
15	1	0.9	5.12	2.56	yes		55	100.00	5500	
15	1	0.9	5.12	2.56	yes	yes	70	50.00	3500	<---

Figure 4. Design spreadsheet for a fictitiously simple application design space consisting of only 12 possible systems. The mission is to search an area A = 10000 and achieve a detect fraction D = 0.99. A base vehicle costs 10, and has a range of 100. The additional cost of better batteries which double the range to 200 is 15. The cost of sensors and processing to implement a completely coordinated search strategy is 30. Three possible mission sensors are available. Equation (4) has been used to calculate the number of elements required, the cost per element, and the total system cost for each of the twelve possible combinations of mission sensor, random or coordinated behavior, and baseline or improved batteries. It is seen that (a) the most cost effective system using the least expensive sensor uses the simplest possible elements, (b) the intermediate sensor justifies the incorporation of the battery upgrade, but not the coordinated search behavior, and (c) using the most expensive sensor justifies both the battery upgrade and the coordinated search behavior.

6. CONCLUSION

The key message of this paper is this: cost effective many-robot system designs may well depend *sensitively* on the interplay of sensor cost and performance levels with mission-specific functional and performance requirements. While the coverage concept may be a useful paradigm for description and analysis, different implementations of real world system systems that provide "coverage" in its various forms will exploit a variety of sensor resources for a variety of fundamentally different purposes, as system designers ruthlessly exploit the opportunities provided by the presence of mission-specific sensors. And, in fact, this is the model for system development provided by natural systems: imaginative opportunism.

The development of useful many-robot systems will, however, require resources at a scale beyond foreseeable technology driven research budgets -- a project must be focused to a single real world application (or a small set of related applications). The ultimate goal of this analysis, simulation, and modeling effort is therefore to identify good candidate real world military and civilian applications, to quantitatively characterize mission requirements, and to develop conceptual designs for systems of intelligent robots that could satisfy them.

7. ACKNOWLEDGEMENTS

This work is supported by the Intelligent Systems Program of the Office of Naval Research, Computer Science Division. The author wishes to acknowledge stimulating and critical discussions with John Bay of Virginia Tech, Dick Blidberg of the UNH Marine Systems Engineering Laboratory, Alan Meyrowitz and Teresa McMullen of ONR, and Lynne Parker, Maja Mataric, and Anita Flynn of the MIT Artificial Intelligence Laboratory.

8. REFERENCES

1. Flynn, A.M. "Gnat Robots (And How They Will Change Robotics)", **Proceedings of the IEEE Micro Robots and Teleoperators Workshop**, Hyannis MA, 9-11 November 1987. Also appeared in **AI Expert**, December 1987, p 34 et seq.

2. New York Times, 1 January 1991, pp 35-39.

3. Freedman, D.H. "Invasion of the Insect Robots", **Discover**, March 1991, pp 42-50.

4. "Ants under Study as Model for Small, Low Cost Robots", **Richmond Times-Dispatch**, 21 June 1992.

5. Drexler, K.E. **Engines of Creation**, New York: Anchor Press Doubleday, 1986.

6. Macedonia, R.M., and B. Sert. "Low Cost Mobility for Robotic Weapons", **AUVS-88, the Fifteenth Annual AUVS Technical Symposium**, San Diego CA, 6-8 June 1988.

7. Megatek Corporation, "CRICKET: A Novel LPI Communication Concept", Report R2017-003-IF-2, 30 September 1978.

8. Brooks, R.A., Maes, P., Mataric, M. J., and More, G. "Lunar Base Construction Robots", **Proceedings of the IEEE International Workshop on Intelligent Robots and Systems - IROS '90**, pp 389-392.

9. Stommel, H. "The Slocum Mission", **Oceanography**, April 1989, pp 22-25

10. Miller, D.P. "Multiple Behavior-Controlled Micro-Robots for Planetary Surface Missions", **Proceedings of the 1990 IEEE International Conference on Systems, Man and Cybernetics**, Los Angeles CA, November 1990, pp 289-292.

11. Bonasso, R.P. "Creature Co-Op: Achieving Robust Remote Operations with a Community of Low-Cost Robots", **Fifth Conference on Artifical Intellgence for Space Applications**, Huntsville AL, May 1990, pp 257-269.
12. Flam, F. "Swarms of Mini-Robots Set to Take on Mars Terrain", **Science**, 18 September 1992, p 1621.

13. Gage, D.W. "Command Control for Many-Robot Systems", **AUVS-92, the Nineteenth Annual AUVS Technical Symposium**, Huntsville AL, 22-24 June 1992.

14. Partridge, B.L., Pitcher, T., Cullen, J.M., and Wilson, J. "The Three-Dimensional Structure of Fish Schools", **Behavioral Ecology and Sociobiology**, vol 6, 1980, pp 277-288.

15. Van Olst, J.C., and Hunter, J.R. "Some Aspects of the Organization of Fish Schools", **Journal of Fisheries Research Board Canada**, vol 27, 1970, pp 1225-1238.

16. Reynolds, C.W. "Flocks, Herds, and Schools: A Distributed Behavioral Model (Proceedings of SIGGRAPH '87)", **Computer Graphics**, vol 21 (4), pp 25-34.

17. Deneubourg, J.L., et al. "The Dynamics of Collective Sorting Robot-Like Ants and Ant-Like Robots", **Proceedings of First International Conference on Simulation of Adaptive Behaviors**, pp 356-363.

18. Franks, N.R. "Army Ants: A Collective Intelligence", **American Scientist**, vol 77, March-April 1989, pp 139-145.

19. Seeley, T.D. "The Honey Bee Colony as a Superorganism", **American Scientist**, vol 77, November-December 1989, pp 546-553.

20. Brooks, R.A. "A Robust Layered Control System for a Mobile Robot", **IEEE Journal of Robotics and Automation**, RA-2, April 1986, pp 14-23.

21. Braitenberg, V. **Vehicles: Experiments in Synthetic Psychology**, Cambridge: MIT Press, 1984.

22. Beni, G., and Wang, J. "Theoretical Problems for the Realization of Distributed Robotic Systems", **Proceedings of the 1991 IEEE International Conference on Robotics and Automation**, Sacramento CA, April 1991, pp 1914-1920.

23. Harmon, S.Y., and Gage, D.W. "Current Technical Research Issues of Autonomous Robots Employed in Combat", **Proceedings of 17th Electronics and Aerospace Conference (EASCON)**, Washington DC, 10-12 September 1984, pp 215-219.

24. Gage, D.W. "Security Considerations for Autonomous Robots", **Proceedings of Symposium on Security and Privacy**, Oakland CA, 22-24 April 1985, pp 224-228. Reprinted in **Computer Security Journal**, vol 6, 1990, pp 95-99.

25. Sugihara, K., and Suzuki, I. "Distributed Motion Coordination of Multiple Mobile Robots", **Proceedings of the 5th IEEE International Symposium on Intelligent Control**, Philadelphia PA, 5-7 September 1990, pp 138-143.

26. Stone, L.D. **Theory of Optimal Search**, Academic Press, New York, 1975.

27. Benkoski, S.J., Monticino, M.G., and Weisinger, J.R. "A Survey of the Search Theory Literature", **Naval Review Logistics**, Volume 38, August 1991, pp 469-494.

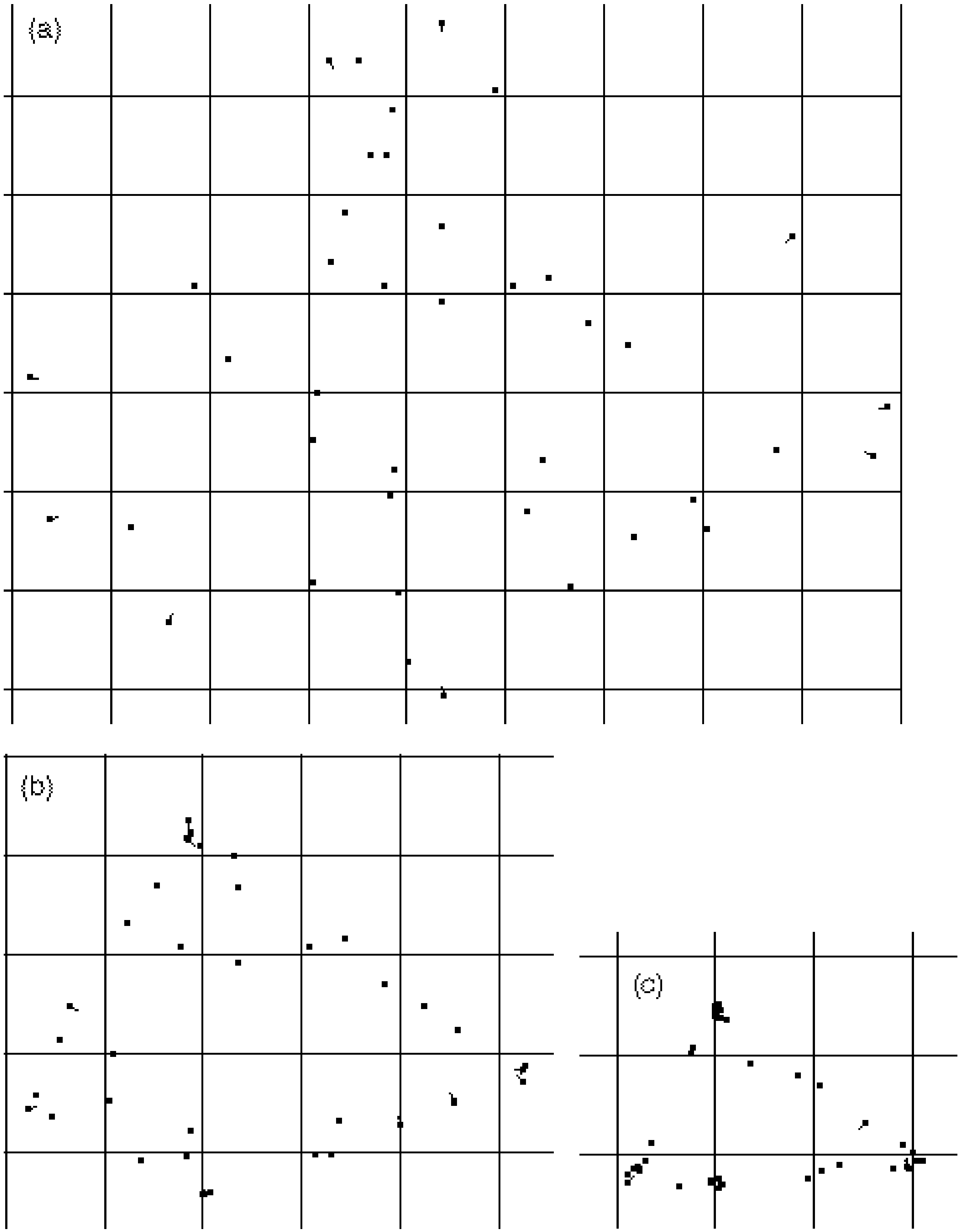
(a)
(b)
(c)

Figure 1. Simulation of "Condensation" algorithm for 40 elements. The line extending from each element indicates its velocity vector, and is not a "tail"; many elements are not moving. (a) First frame after the condense algorithm has been initiated; the element positions reflect random initial conditions. (b) 23 frames after the first frame shown as (a); the group diameter is 42% smaller than when condensation began. (c) 27 frames after the frame shown as (b); note the concentrations of elements at the "corners" of the configuration

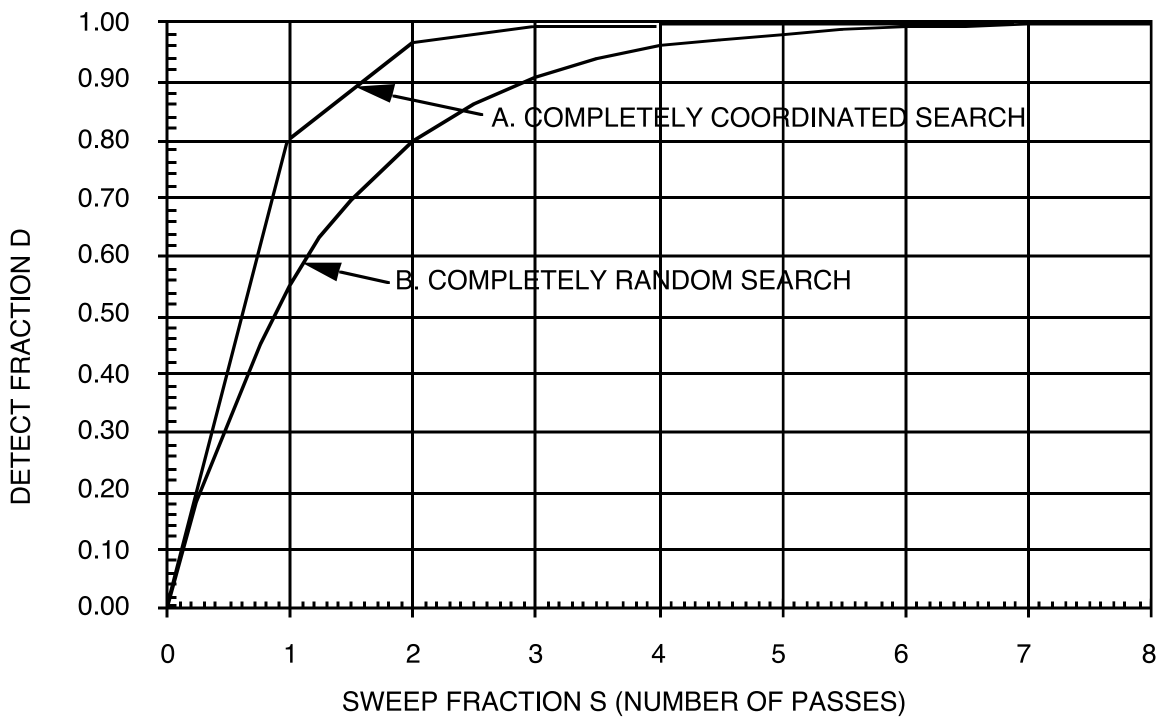

Figure 2. Detect Fraction (D) as a function of Sweep Fraction (S) for sensor probability of detection (p) equal to 0.8. (a) Completely coordinated search strategy, equation (1). (b) Completely random search strategy, equation (2).

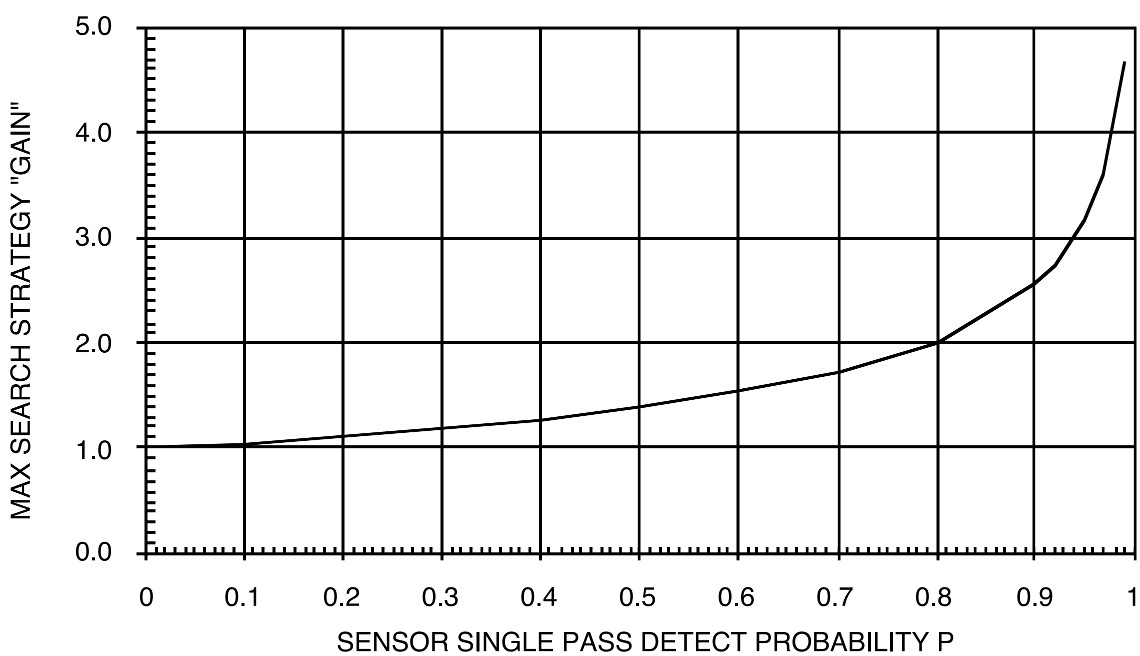

Figure 3. Maximum possible search strategy gain (G) as function of the sensor detection probability (p), equation (3).