Ultra-low complexity control mechanisms for sensor networks and robotic swarms

Authors: Matthias Scheutz and Peter Bauer
Date: Jan. 2013
From: International Journal of New Computer Architectures and Their Applications(Vol. 3, Issue 1)
Publisher: The Society of Digital Information and Wireless Communications
Document Type: Report
Length: 11,410 words
Lexile Measure: 1640L

Abstract:

Biologically inspired swarms of autonomous robots have been used successfully in a variety of robotic applications ranging from various kinds of ground-based robots, to unmanned aerial vehicles. Typically, all of these systems use digital communications among swarm agents to implement their behavioral rules (e.g., because they need to exchange information about the location of agents).

In this paper, we propose a general control architecture for ultra-low complexity robotic swarms that can be fully implemented in analog hardware and does not require digital communication for any part of the swarm coordination. We demonstrate the versatility and effectiveness of the proposed mechanisms both in simulations and on a robotic swarm platform for a variety of applications, ranging from area coverage, to target tracking, target interception or target enclosure, to active exploration and target finding, among others. The proposed system is extremely simple, robust, and scales well. It allows for homogeneous and heterogeneous swarms and has been successfully applied in several physical instantiations.

KEYWORDS

Swarm robotics, ultra-low complexity swarms

Full Text:

1. INTRODUCTION

Over the last decade, swarm intelligence [3] has become an interesting alternative to standard centralized and distributed control approaches (e.g., [23, 26, 30, 32, 33, 34]) for solving a variety of multi-agent coordination problems. Proposed approaches range from using only local rules (e.g., [11]), to "digital pheromones" [2, 4], to sensor fusion [9] and the self-deployment of sensors [13], to forming formation [5, 12], to chemical plume detection and tracking [1,8], and many others.

Different from centralized control where one or more complex agents determine all actions that the other agents have to perform (and thus communicate action sequences to them, which they will simply carry out), swarm systems do not have such a central source of behavior coordination. Rather, global behavior emerges from the interaction of the swarm agents, whose behavior is usually governed by simple rules. These rules typically use information about an agent's immediate neighbors (e.g., Reynold's three rules "flock centering", "obstacle avoidance", and "velocity matching" [6]). Based on assumptions about such simple interactions, several important theoretical results have been proved recently about properties like collision avoidance or stability in swarms systems (e.g, [15, 16, 17, 19, 20].

Often times, swarm rules assume that information about the exact position and orientation of a swarm agent is available (e.g., for an agent to be able to compute the distance and heading of all its neighbors). While it is straightforward to obtain the exact position of an agent in simulations of swarm systems (and thus properties like the distance between any two agents), in a practical settings this information requires special sensory and functional capacities, in particular, digital communication (e.g., GPS on each agent and digital communication to relay the GPS data to other agents that require them). For example, agents in a swarm system that is based on "digital pheromones" [2, 4] will need to maintain a global map of pheromone positions that is shared among all agents and updated through communication. While this approach of using digital communication for behavior coordination is useful in some scenarios, it has several problems, especially for large physical swarms. For one, it does not scale well with the size of the swarm. Moreover, digital communication can fail and failures may cause deterioration of the swarm, which needs to be explicitly addressed. This in turn leads to additional control and hardware mechanisms, increasing the complexity of each swarm agent. In general, we believe that assumptions about the availability of information from swarm agents as are often made in simulation studies are idealized and difficult to meet in practical implementation. In particular, it might not be possible to get exact distance readings for all agents within a given neighborhood (e.g., because of occlusion effects, or simply because sensors that could provide that information are too expensive or complex to be used in the agent implementation, lack of GPS, etc.). Moreover, it might not be possible to determine the overall goal direction (e.g., because the goal cannot be sensed at a distance). Finally, theoretical investigations typically limit swarms to homogeneous groups, while it might be advantageous to use heterogeneous groups, e.g., because they give rise to a more robust system, improve overall task performance, or require fewer resources.

In this paper, we take a different approach that derives its motivation directly from the organizational and functional principles of biological multi-agent systems: (1) labor is divided among many very simple, autonomous, expendable individual agents with simple control systems, which (2) not only guarantees the reliability of individual agents, but also a high level of fault tolerance of the overall system. Most importantly, (3) the overall system behavior "emerges" from the interactions of the individual agents and is accomplished in a distributed, collaborative fashion. Thus, rather then adding mechanisms to increase the reliability of communication to guarantee swarm performance, we propose to abandon digital communication altogether and use a novel, ultra-low complexity navigation and resource allocation concept that is exclusively based on simple radio beacons carried by each swarm agent. Different from other systems, the proposed principle can be implemented with extremely simple hardware and does not require the existence of a digital communication network to perform its functions. In principle, there are two types of beacons that every agent carries: an attractive beacon for resource allocation (moving swarm agents to where they are needed) and a repulsive beacon to control agent distribution and density in order to avoid collisions and control sampling density. In the most general framework, all entities carrying these two beacons are assumed to be mobile, even though the case of some beacons being stationary also generates interesting applications.

The navigation principle described in this paper has been implemented for a number of different application scenarios and performance data presented in this paper is based on these simulations. Using extensive simulations, we are able to demonstrate that (1) the system functions as expected and can achieve a great variety of different coordination tasks, (2) scales up (i.e., it works with an arbitrary number of agents), and (3) is robust to disturbances (adapts to changes in the number of agents, their positions and configurations). Moreover, several of the simulated applications have been implemented in a ground-based robotic prototype swarm system consisting of simple rovers with embedded controllers in order to validate the simulation results in a real environment, testing assumptions on models, uncertainties, parasitic effects, and others.

The paper is organized as follows: we start with a description of the proposed swarm system, providing details of the control mechanisms, on how swarm agents can be realized in hardware, and the robotic evaluation platforms we have used. We then describe some of the interesting system properties and extensions and move on to demonstrating the utility of the proposed mechanisms in four different application scenarios. Reporting results from computer simulations and test runs on robots, we demonstrate the viability and utility of the control mechanisms for real-world applications.

2. ULTRA-LOW COMPLEX ITY CONTROL FOR ROBOTIC SWARMS

We are interested in developing an ultra-low complexity control mechanisms for swarms that can be easily realized in a physical robotic system without the need for digital communication. The control mechanisms should allow swarm agents to explore the environment, follow signals and signatures of interest, and possibly relay information of interest back to a home base. Moreover, it should allow for swarms of arbitrary size. We start with a description of the two types of beacons that every unit carries: an attractive beacon for resource allocation (moving autonomous agents to where they are needed) and a repulsive beacon to control agent distribution and density in order to avoid collisions and control sampling density.

2.1 The Basic Navigation Concept

Each swarm agent is equipped with two types of beacons, a collision avoidance beacon col (to repel agents from each other and distribute them) and a target attraction beacon tar that is only activated when agents detect a target (whatever that target may be). The collision avoidance beacons are always on and each agent is equipped with a stereo antenna/receiver pair to detect collision avoidance beacons of other agents (and their direction). In the same way, each agent uses the antenna/receiver pair to detect the target attraction beacons of other agents. Note that target attraction beacons effectively translate a signal from a target (regardless of how it was detected) into a different modality (i.e., an on/off radio beacon) to extend the range of its detectability. In contrast to target beacon receivers, the collision avoidance receivers can extract the approximate distance of the source using the received signal strength. This allows the collision avoidance algorithm to react only to agents within a certain circle of radius p (the "repulsion radius") in free space. p effectively is an agent's collision avoidance range and represents the distance an agent must keep between itself and its neighbors to leave enough space to turn. Therefore, p is dependent on the agent's minimal turning radius t, which, in turn, is typically dependent on the agent's speed.

For the control algorithm, we define: = [L.sub.y,i] [[summation].sup.n.sub.j=1][A.sub.y]/([parallel] [[X.sub.j].bar] - [[x.sub.i].bar] [[parallel].sup.2.sub.2]) where [I.sub.y,i] is a measure proportional to the received power of beacon type y at agent i at location [x.sub.i] with y [member of] {col, tar}, and [A.sub.y] is the transmit power of type y beacons. (There are a total of n agents, and all beacons of the same type have the same power.) Using the directional sensitivity of two sideway looking directional antennas, we can find the following signal intensity for left and right looking antennas of each of the two modalities:

[MATHEMATICAL EXPRESSION NOT REPRODUCIBLE IN ASCII]

with [[GAMMA].sub.col] = {j| [parallel] [[x.sub.j].bar] - [[x.sub.i].bar] [[parallel].sub.2] < [rho]}, [[GAMMA].sub.tar] = {1, ***, n}, [[[eta].sub.i].bar] being the right normal vector to the speed vector of agent i in the plane of operation (i.e., either on the ground or in the flight plane), and f (x, [eta]) being the directional sensitivity function of the antenna, where x is the vector from receiver to source and [[eta].bar] is the direction of highest sensitivity of the antenna.

In the case of y = col, the summation for the left and right antenna signal intensity [L.sub.i] and [R.sub.i] respectively are taken only over those agents j that satisfy [parallel] [x.sub.j] - [x.sub.i] [[parallel].sub.2] < [rho]. This requires certain provisions in the modulation scheme that allow the agent to distinguish each collision beacon. (In the attraction beacon receivers, distinguishing between different agent beacons is not necessary and the above summation can actually be done by the antenna itself rather than in digital hardware.)

The decision for the turn direction requires two directional antennas on each side of the agent facing in opposite directions ([[eta].bar], and -[[eta].bar]), perpendicular to the agent speed vector. Since the turning radius of the agent is assumed independent of the direction (left and right), a simple intensity comparison between left and right directional antenna will allow to derive the new heading of the agents, which is either "turn left" or "turn right". Define the intensity sum and difference between the antenna pairs as: [L.sub.y,i] + [R.sub.y,i] = [S.sub.y,i], [L.sub.y,i] - [R.sub.y,i] = [D.sub.y,i], y [member of] {col, tar}, i = 1, ***, n, where [S.sub.y,i] and [D.sub.y,i] denote sum and difference of left and right antenna signal strength of modality y at agent i. The agent control algorithm is thus as follows:

```
while true do
  if [S.sub.col,i] > 0 then
    if [D.sub.col,i] > 0 then
```

```
      turn right
    else
      turn left
    end if
  else if [S.sub.tar,i] > threshold then
    if [D.sub.tar,i] > 0 then
      turn left
    else
      turn right
    end if
  else
    fly straight
  end if
end while
```

Note that it is possible to simplify the above algorithm even more if the assumption is dropped that the repulsive beacons of all agents be distinguishable. Then the same summation can be done for the repulsive beacons as is done for the attractive beacons, without the need to use n different repulsive beacons. The downside of this simplification is that it is now possible that an agent will sporadically make wrong decisions about where to turn. Suppose there are two agents to left of agent A at a distance of 100 that have their repulsive beacon on and A's repulsion range is p = 80. Then the intensity reading on A's antenna will (wrongly) suggest that there is an agent within repulsion range and A will turn to the right even though there was not impending collision. This situation can be problematic, at least in principle, when there are lots of agents on one side whose summed repulsive beacon intensity make A turn in the opposite direction right into the trajectory of another agent whose is also within collision range, but whose beacon intensity is lower than that of the group of agents on the left. While this situation can lead to collisions in principle, it is practically much less problematic, for two reasons: first, because agents are normally separated by a distance of at least p and beacon intensities falls of with the at least the square of the distance, the influence of far away agents is negligible and only a small number of close agents will determine A's behavior. And second, agents do not have to have their repulsive beacon on all the time, but can rather send pulses at a certain frequency (possibly with a slight random component added). Then the probability that two or more pulses will occur at the same time can be kept very low, and even if they should co-occur, a short period of time later they will be spaced again and the agent can make the right decision (note that in this case we need to add a memory component in the control system that for a short period of time stores the last beacon intensity).

One possible direct implementation of the algorithm in digital hardware is shown in Fig. 1. The hardware implementation consists of two power complementary antennas that are used for both, the target and the collision beacons. The left and right collision beacon receivers demodulate the unique collision beacon signals and provide at the output the detected signal strength for each of the n agents. Since the collision beacon strength of all agents are equal, and in free space the power drops with the square of the distance (on the ground the exponent is typically anywhere between 2.5 and 4), one can approximately determine the threshold for the received collision beacon signal that corresponds to the critical distance p. This threshold is implemented in the two threshold filters, which pass the collision beacon signal of agent i only if it satisfies the critical distance (received power) requirement. All collision beacon contributions that pass the threshold test are then added for the left and right side, and in a second step subtracted to find which side produces a stronger signal. This difference is then converted into a left or right turn signal using a (signum) hard limiter nonlinearity. The target beacon signals are compared in a similar way. However, in this case, all agent contributions are used and are summed up by the antennas themselves. If the left plus right antenna signal strength passes the given threshold, the left and right turn signal is generated in the same way as in the collision beacon case. The channel selector is a priority selector, that selects channel 0 with highest priority (collision beacon generated command signals) if it is enabled, then channel 1 with second highest priority (target beacon generated command signals) and finally with lowest priority channel 2 which is the alternating left right signal. This signal results (after the compensator) in a constant heading. From the schematic it is clear that the control scheme can be easily realized as a neural network with perceptrons: input units are used to represent the signals coming form the antennas and are connected to hidden units (the "summation nodes") that perform summation and thresholding, which, in turn, are connected to the output units that implement the channel selector. Note that this figure only shows control mechanisms in the 2-D plane and that altitude control (e.g., for the 3D case of UAV agents) and the thresholding circuit for activating the target beacon are not shown.

The continuous-time motion model of a swarm agent based on the above control algorithm and the discrete-time motion model (using delta-operator-based discretization) are given in Figure 2.

2.2 Realization and Variations

Some applications that we will discuss in the next sections may require multiple building blocks of the type shown in Fig. 1 and thus a selection mechanism that chooses one of the available outputs provided by the multiple hardware blocks. Both selection mechanism and beacon ranges are application-dependent. Note that it is possible to integrate information from active range finder sensors (e.g., sonar or laser sensors) for obstacle avoidance in the control system as part of the col beacon circuit (for example, for UGVs that need to avoid road-side obstacles). In that case, no beacons from other agents need to be received (rather signals originate from and are measured in the active sensor component of the agent), hence no col receivers are needed and the the sensor signals can be directly fed into the control circuit. Similarly, agents can be directed in particular regions via virtual attractive sensors (e.g., GPS locations of a target area, which are converted into directional vectors based on the agent's current GPS location), which are fed directly into the tar control circuit. None of these additional mechanisms (for additional obstacle avoidance and target-oriented navigation) require a change of the functionality of the basic navigation system. Consequently, basic properties of the agent system (as discussed in the next section) are preserved across different configurations of sensor suites. That way it is possible to use multiple distinguishable attractive and repulsive beacons in conjunction with multiple non-beacon-based inputs (also designated "attractive" or "repulsive") in order to create agents with quite sophisticated behavior (including selective collision avoidance with objects of a particular type, target tracking and following in obstacle environments, etc.) based on the very same control mechanism.

2.3 Implementation in Robotic Agents

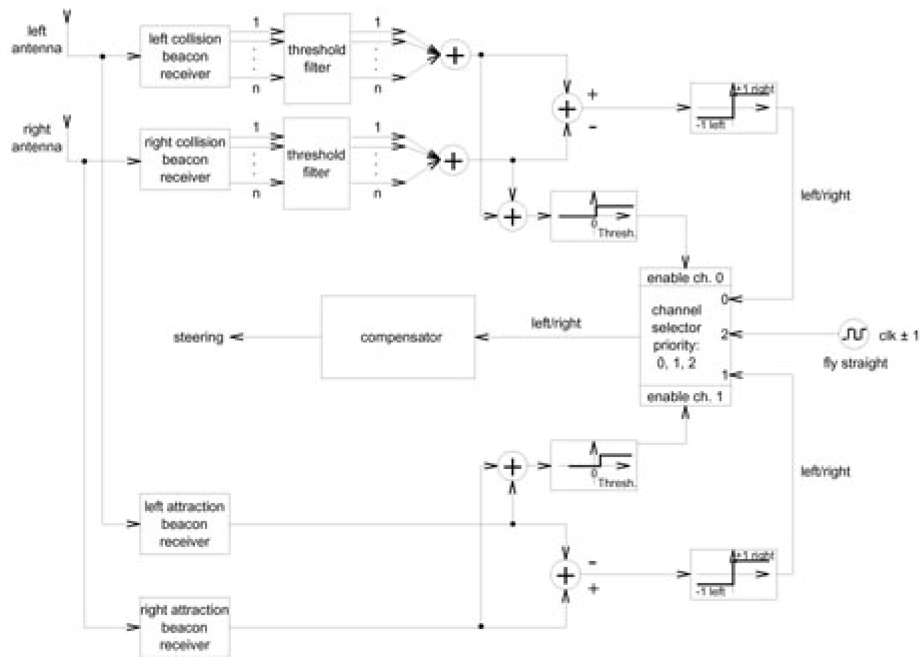

Figure 1: The implementation of the agent control algorithm in hardware.

The entire robotic platform has two major components: mobile swarm agents and stationary waypoint beacons. We are going to describe both components in some detail.

Mobile Swarm agents: Each swarm agent consists of a modular 4 wheel steel based platform, generated from the so called VEX robotic kit. This platform provides the chassis for the vehicle, the power train unit (4 wheel drive gearing, motors, motor controllers, etc.), tactile sensors, and the processor. In the described application, the VEX processor was used only to translate the direction information computed in the TelosB sensor nodes to drive motor information. Each platform is equipped with two side looking TelosB wireless nodes. Each TelosB node uses the Zigbee protocol for communication. (1) The geometric arrangement of the two nodes is symmetric, with the PCB integrated antenna facing away from the vehicle at an angle of approximately 90 degrees. In order to get better separation (gain differences) between the two antennas, additional metal shields are used between the two nodes. Each TelosB is powered by a 7.2V Lithium Polymer battery followed by a DC to DC converter that produces a 3.3V output voltage. The same battery powers the Vex processor and the motors. It is therefore guaranteed that the TelosB processors do not run out of power, i.e. motor power is lost first.

The TelosB implementation of the proposed ultra low complexity scheme is done using a master slave arrangement: the left facing TelosB node represents the master, while the right facing TelosB node acts as the slave that forwards its RSSI information to the master. The master will then determine the difference between right and left RSSI and make the decision to turn right or left depending on the type (attractive or repelling beacon) of packet that was received. The comparison is made approximately every 300ms-500ms, which is also the beacon firing period for all agents and waypoints. Master and slave also serve as the repel (and in certain instantiations of the scheme as the attract) beacon. Since both, master and slave are shielded from each other, the slave repeats the master beacon after approximately 150ms in order to get a approximately symmetric range to both sides of the swarm agent. (In fact, both, master and slave use the same ID when sending the beacon packet, so they are indistinguishable by other agents.) Typically, the repel radius is set at about 1-3 meters, while the attract radius depends on the application. If a task calls for attaching sensors to the swarm agent, then typically the master node is interfaced with a plug-in sensor board (Easysen SBT30 board).

Note that even though Zigbee is a digital communication protocol, it was not used for digital communication in the evaluation platform, but rather the intensity of the Zigbee beacons were used to determine the distance between agents as described in the algorithm. We simply used Zigbee because the TelosB nodes were readily available (compared to building the analog beacon circuit as proposed from scratch). In fact, the Zigbee implementation of the proposed analog control scheme already shows that a digital implementation does not scale well which is in stark contrast to the analog implementation.

Stationary Waypoint: A stationary waypoint is implemented as a single TelosB node with a long range antenna. Typical beacon range is between 20 and 100 meters, depending on the power levels used and the altitude of the antenna. Waypoint beacon firing rates are typically 300ms-500ms. Waypoints can optionally be sending a small range repel beacon also, making sure that no collisions with the waypoint occur. However in practice, this is often not necessary since in a finite state machine implementation, the agent will look for a new waypoint (N+1) after it has come sufficiently close to the waypoint (N).

$$\underline{\dot{V}}(t) = \begin{bmatrix} 0 & \beta\,\mathrm{sgn}(R-L) \\ -\beta\,\mathrm{sgn}(R-L) & 0 \end{bmatrix} \cdot \alpha \cdot \underline{V}(t)$$

$$\begin{aligned} \underline{V}[(n+1)T] &= \underline{V}(nT) + T \cdot \begin{bmatrix} 0 & \beta\,\mathrm{sgn}(R-L) \\ -\beta\,\mathrm{sgn}(R-L) & 0 \end{bmatrix} \cdot \alpha \cdot \underline{V}(nT) \\ &= \begin{bmatrix} 1 & \beta\alpha T\,\mathrm{sgn}(R-L) \\ -\beta\alpha T\,\mathrm{sgn}(R-L) & 1 \end{bmatrix} \cdot \underline{V}(nT) \end{aligned}$$

Figure 2: The continuous-time motion model of a swarm agent based on the above control algorithm (top) and the discrete-time motion model (using delta-operator-based discretization (bottom) with α controlling the turn radius and T being the update (sampling) time. The coefficient β takes a value of 1 for an attractive beacon and -1 for the case of a repel beacon. Also the speed vector $\underline{V}(nT)$, $\underline{V}(t) \in \mathbb{R}^2$, and $\mathrm{sgn}(x)$ is the *signum* functions, $\mathrm{sgn}(x): \mathbb{R} \rightarrow \{-1, +1\}$. R and L are receive signal strength of the right and left looking antenna respectively.

3. SYSTEM PROPERTIES AND EXTENSIONS

As mentioned, the proposed system allows large numbers of unmanned agents (UAVs, UGVs, etc.) to self-organize and jointly achieve tasks that involve navigation for positioning, tracking, and interception of other moving objects in the air and on the ground. Several system properties are of interest: (1) dense hex-grid coverage and formation, (2) collision-free navigation, (3) scalability and reliability, (4) adhoc sensor network and routing of information, and (5) mechanisms for deployment and recollection of agents. (2)

Currently, we do not yet have a fully developed framework that would allow us to report formal results for confirming the above properties which we have observed in simulations and the robotic platform. However, we believe that it is still valuable to briefly mention a possible approach towards obtaining formal results. In particular, we expect an analysis based on the dynamics of the swarm through the use of a discretized space-time system to most successful that carries state information for each individual agent (location and heading only). Even though the swarm (at non-zero agent speed) cannot have an equilibrium (in terms of the system states), we can still analyze the dynamic swarm properties using stability theory based on attractive and positive invariant sets. So, in the case of a single attractive waypoint beacon and a set of identical mobile agents, this stability problem would translate into having all mobile agents enter in finite time a region centered at the attractive beacon, with the requirement that agents will not leave this region any time after that. In order to evaluate the distance functions of agents from each other (and thus capturing collisions and near-collisions) a performance index that is based on the sum of pairwise distances of agents from each other can be used, which will provide conditions under which the state trajectories can be bounded over time (including information on the bound in 2-D space) and allow us to evaluate how well agents avoid collisions. Another possibility might be to combine the above with probabilistic and graph-theoretic methods. The former can be used to obtain probabilistic system performance measures, while the latter can be used in discretized environments to obtain results regarding collision avoidance and reachability of targets under various environmental constraints.

In the following, we will briefly de scribe some interesting properties of the swarm system.

3.1 Dense Hex-Grid Coverage and Formations

For efficient sensing and sampling, agents (UGVs or UAVs at the same altitude) must form a dense cover of subregions of the 2D plane, the density of which is determined by the agents' repulsion range p (top in Fig. 3; the dashed circles depict this radius p). Given one attractive beacon in the center of an area, for example, agents will automatically arrange (and continuously re-arrange) themselves in the vicinity of the beacon such that agents are outside each other's repulsion ranges. Specifically, since agents within each other's repulsion ranges move away from each other, while being attracted to the center of the region when they are not within each other's repulsion ranges, we will get a stable oscillation (i.e., a pattern of agents moving in and out of each other's repulsion ranges while staying in the same overall area). Simulations demonstrate that this behavior of agents leads to an emergent arrangement of agents on a hexagonal grid of approximately p grid length (i.e., the shortest distance between two agents before they ignore the attractive beacon and turn away from each other based on their navigation control system, see bottom in Fig. 3). Note that this is the tightest possible packing of circles in the plane, and thus the optimal arrangement of agents with circular non-overlapping repulsion regions. Note that because agents are constantly moving, they will be in an out of each others repulsion range, hence the hexagonal formation will be necessarily dynamic and approximate. Yet, for small T compared to p, we expect the pattern to be sufficiently stable.

3.2 Homogeneous vs Heterogeneous Swarms

The proposed control scheme naturally allows for homogeneous and heterogeneous swarms, where the difference in agents comes from differences in [rho] (possibly based on differences in [tau]). In heterogeneous swarms, interesting patterns of coordination can emerge based on the frequency with which attractive beacons. Figure 4, for example, shows a swarm consisting of three kinds of agents ("brown", "green", and "blue") with three different repulsion ranges (brown being largest and blue smallest, green in between). The swarms is attracted to a cloud of particles, because agents will turn on their attractive beacons whenever they encounter a particle. The top row shows the agent shortly after they encountered the particle cloud with mostly green agents being in the center (the left column shows an overview of the environment, the right column shows a zoomed version of a subarea indicated with a square on the left). Very quickly, the blue agents start to move towards the center, with the green agents grouping around them, and the new incoming brown agents (moving in from the left) forming a ring around the green agents. The results is a dynamically stable pattern of concentric circles (bottom row). This pattern is very robust and emerges due to the asymmetric interactions between agents with different repulsion ranges. Specifically, agents with a smaller repulsion range (e.g., blue agents) encountering agents with a larger repulsion range (e.g., green agents) will cause the agents with the larger repulsion range (i.e., green agents) to turn away without themselves having to turn away (because even though they are within the repulsion range of the other agents, the other agents are not within their repulsion range). This asymmetric penetration of the agents' repulsion ranges thus allows agents with smaller repulsion ranges to move towards the target location, while agents with larger repulsion ranges will have to remain at a distance. Ultimately, this asymmetry gives rise to the emergence of the agent distribution forming concentric circle around the source of the attraction.

3.3 Collision-free Navigation

We conjecture that if repulsion ranges are chosen carefully such that the minimal turning radius [tau] < [rho]/4-[delta], where 8 is some safety distance, then it is always possible for agents to avoid collisions. In the worst case, they will be able to repeat a circular pattern of radius [tau] in a region within their repulsion range [rho]. Specifically, as shown on the right in Fig. 3, a complete enclosed agent can still safely turn away from a set of six surrounding agents, all of which have penetrated the repulsion range of the enclosed agent (note that these agents also have a safe place to turn within the enclosed agent's repulsion range).

Formally establishing collision-free avoidance is currently an interesting, open problem for the proposed swarm system. Specifically, it would be interesting to isolation conditions for p and [tau] such that for given a fixed minimum speed [v.sub.o] of all agents it is guaranteed that collisions cannot occur for an arbitrary number of agents as long as all agents start out from a "safe" position with non-overlapping repulsion ranges.

3.4 Scalability and Reliability

One of the interesting properties of most swarm systems is that they tend to "scale up", i.e., new agents can be simply added to a system without usually negatively impacting the performance of other agents. For example, if a system S consisting of k agents has achieved insufficient coverage of an area A (i.e., only p * A for 0 < p < 1 is covered), then perfect coverage of A can be achieved by adding at least (1 - p) x A/2[pi] x [[rho].sup.2]) new agents of the same type. Similarly, if a system S consisting of k agents has achieved coverage of an area A by covering A + d (where d is the excess area covered), then removing an agent (e.g., because it ran out of fuel or was destroyed) will either still cover A (if d > 2[pi] x [[rho].sup.2], i.e., the area covered by one agent) or A will occasionally not be entirely covered.

3.5 Adhoc Sensor Network and Routing of Information

If agents are equipped with wireless communication devices, they can automatically form an adhoc wireless network as soon as they come sufficiently close to each other, assuming that the communication range Y > p (otherwise they could only communicate when they are performing evasive maneuvers).

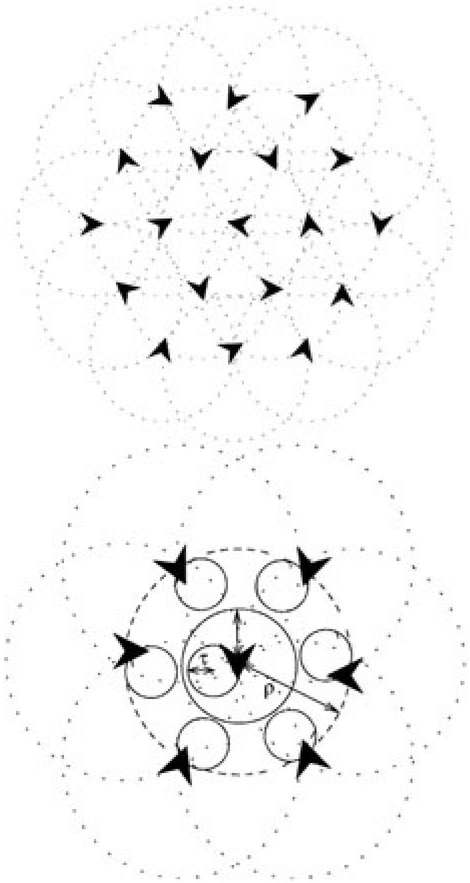

Figure 3: Coverage of area by agents (top) and collision-avoidance (bottom). Dashed circles depict the repulsion range ρ. τ is the minimal turning radius.

3.6 Mechanisms for Deployment and Recollection of Agents

Mechanisms for automatically deploying and recollecting agents are an important part of an agent system. For deployment, agents will generally have to be oriented in the expected target direction. E.g., it is possible to make agents follow particular trajectories along "nav points" based on a sequence of attractive beacons that are subsequently ignored. For example, suppose agents have to patrol k areas [A.sub.1],[A.sub.2], ..., [A.sub.k] in sequence, then by deploying different attractive beacons [B.sub.1], [B.sub.2], ... , [B.sub.k] in each area (e.g., "shooting" a beacon in the area or dropping it by aircraft), the agents control system can be modified such that after having encountered beacon Bt (at sufficient strength), Bt will be ignored, and [B.sub.i+1] (for i< k) will become attractive (in this case, only one beacon is attractive at any time). As a consequence, agents will visit each area At in sequence until they detect a target, whose attractive beacon temporarily supersedes any attractions from beacons Bt. For recollection, a similar mechanism is possible: a special "recollection beacon" R is activated, which causes agents to ignore all other beacons and return to the home base.

A simple mechanism using different types of beacons can be used for automatically deploying and recollecting agents such that human operators can easily influence the overall behavior of the agent system (e.g., by selectively activating beacons of subsets of agents) without having to worry about the details of navigation. The same idea can also be used for route programming and executing branching operations along the way point trajectory.

4. APPLICATIONS

We now consider four (of many possible) applications of the proposed swarm system. All four applications have been implemented in in our distributed parallel agent-based simulation and experimentation environment SWAGES [36, 37], and the first two have also been implemented and tested outdoors on real robots.

The first two applications can be performed either by ground and aerial robots individually (we describe them for unmanned aerial vehicles in simulation, and for unmanned ground vehicles for the robotic swarm here). The third demonstrates that the proposed principle works equally well for the coordination for mixed UGV-UAV systems and the fourth show how evidence filters can be easily integrated into the swarm to allow for a problem-focused search, demonstrating elevated swarm intelligence.

Overall, all systems show better performance while reducing the resources requirements than other low-cost applications (e.g., [43, 8, 44, 45]).

4.1 Detection, Tracking, and Reporting of Ground or Airborne Objects and Substances

The task is to detect ground or airborne objects or substances, report their position and track them (in case they are moving). This task can take many different forms depending on what the objects and substances are. For example, UGVs could track and locate communication signals, or detect and track chemical substances in the air. Similarly, UAVs could locate and track moving targets on the ground, or detect nuclear substances in a chemical cloud and determine its extension. Here, we focus on a UAV system that can determine the boundaries of the distribution of a large number of target objects or substances (e.g., chemicals or radioactive substance in the air, individuals on the ground, etc.), which cannot be sensed or identified at a distance.

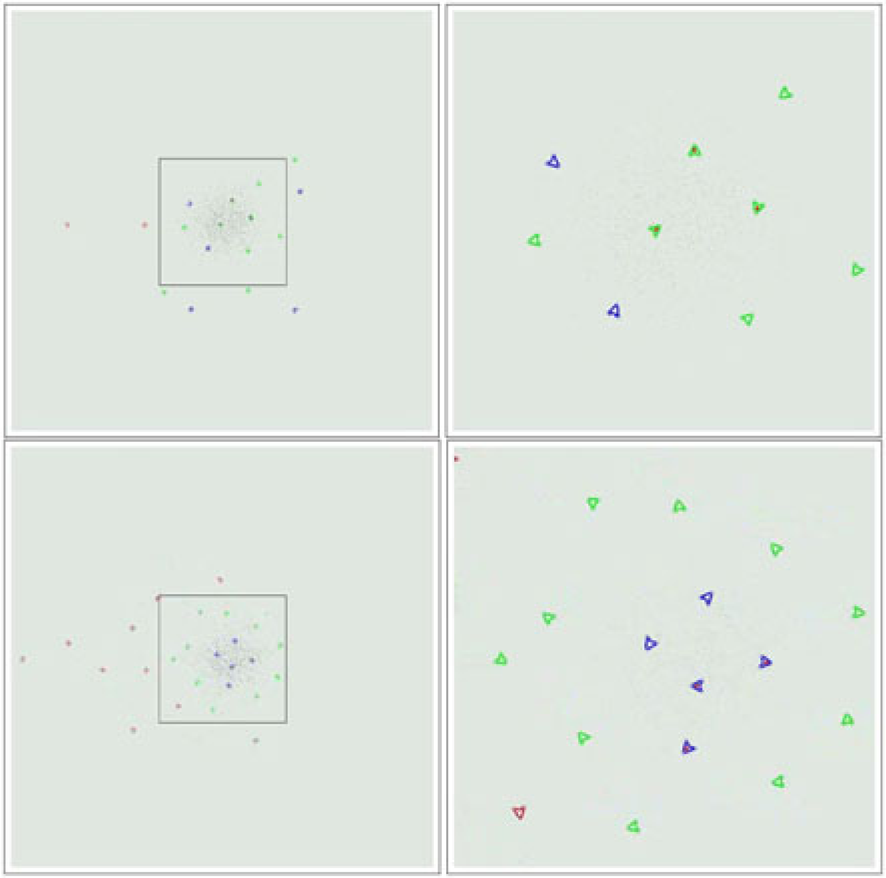

Figure 4: Agents coming to the target area (top) and forming concentric circles based on their collision avoidance range (bottom), see text for more details. The screen shots in the right column show zoomed versions of the quadratic area indicated in the screen shots in the left column.

Fig. 5 (A) through (F) shows various phases and states of the "information gathering task". The targets are indicated by gray points--(D) through (F) depict a magnified view of the targets and their boundary indicated by the polygon. The agent swarm here consists of two kinds of UAVs, those with large-range communication ("red" and "blue"), call them "reporters", and those without ("brown"), call them "workers". Both kinds have short-range communication links (e.g., Zigbee), GPS, and sensors to detect the targets (e.g., soldiers or vehicles on the ground). Initially, a certain number of UAVs of each kind is sent in the direction in which targets are expected (A) (2 reporters and 15 workers here). Lines between two UAVs depict established wireless links (which are established whenever the UAVs come within wireless communication range). As soon as one UAV detects a target, it turns on its attractive beacon and attracts the others to the area (B) for further inspection (and possibly corroboration of the detected information). Note that no information has been transmitted to the home base yet, as the adhoc network is still forming and no "reporter" is part of the network yet. (C) shows the state shortly after the "red" reporter joined the adhoc network of workers forming in the target area, and starts sending information about detected targets back to the home base. Note that other workers can as part of the network forwarding process integrate their sensory data to corroborate the information (e.g., using Bayesian belief updates or simply by keeping previously routed packets with information about close-by locations in a cache to which current sensory information can be compared). The red circles in (D) depict the reported locations of sensed targets and the red polygon indicates the extension of the targets based on the reported measurements (for ease of comparison, the actual locations of the targets are superimposed as gray dots and their extension is indicated by the gray polygon). As can be seen, the center of the target area has been already determined and communicated by the system. (E) then shows the state of the system after about 500 simulation cycles. A dense, dynamically changing ad hoc network has formed with reporters in the center to increase throughput (this is a robust emergent effect of the proposed navigation mechanism). (F) shows again the reported target locations superimposed on the actual ones. Green circles (in addition to red and blue ones) show locations confirmed by both reporters. As can be seen, many individual targets and their overall extension has been determined at this point. Note that none of the demonstrations require central supervisory control to instruct individual UAVs where to move to. Rather, UAVs organize themselves around the targets and will follow them regardless of whether they are stationary or move. Coverage depends solely on the number and distribution of targets, the number of available swarm agents, and the parameters set for the repulsive beacons, which determine the distance swarm agents keep from each other. Some results of our investigations to date are reported in [1].

Note that the dynamic nature of the UAV network and the potentially high rate of changes in positions of UAVs present challenges for networking topologies, protocols, and algorithms. Given the limited communication range of UAVs, any route between any two points can change frequently and unpredictably, and may thus not be available at all from one moment to the next. This means that existing routes have to be periodically checked and new routes have to be discovered if established routes have ceased to exist, possibly requiring agents temporarily to store packets to minimize information loss. Critically, the nature of the connectivity of the network over time will essentially depend on the motion of the agents given that the wireless communication range is limited. Hence, networking tasks (such as establishing communication between agents, discovering and using routes, etc.) cannot be investigated in isolation of the control mechanisms that establish the agents' navigation strategies.

The simulations demonstrate key results similar to results we have obtained previously [1]: (1) the system is capable of finding and tracking targets, (2) the number of required agents will depend on the extension of the target cluster and the repulsion range p, and (3) the average network connectivity is sufficient for fast detection and determination of the extension of the target area.

We would also like to point briefly to the utility of using heterogeneous swarms for these kinds of detection, tracking, and reporting tasks. As mentioned before, heterogeneous swarms consist of agents with different repulsion ranges [rho]. To compare a homogeneous and an heterogeneous swarm for the above task, we let homogeneous agents have [rho] = 150 and agents in heterogeneous swarms have [rho] = 170 and p = 130. Both homogeneous and heterogeneous swarms are capable of finding and tracking targets. Fig. 6 shows the percentage of the actual target area reported by each swarm as a function of simulation cycles for homogeneous (top) and heterogeneous (bottom) units for nine different combinations of reporters (1 to 3) and workers (15, 20, and 25). As can be seen the heterogeneous configurations do generally better than the homogeneous configurations (we find this effect in all other simulations as well). The advantage of different p values is that units with smaller p values can "penetrate" tight arrangements of units with larger p values and thus move from the outside into the center of a network (this is useful for reporters as most information accumulates in the center). In general, heterogeneous swarms will form concentric arrangements of agents based on increasing [rho] values, so that a tighter coverage will be obtained in the center and a looser coverage at the perimeter of the target area. This allows for coverage of large areas with fewer agents than in the homogeneous case.

Robotic implementation: The task of detecting contaminants was implemented by using a homogeneous swarm of VEX ground vehicles. The theater of operation was a university parking lot and the substance to be detected was water on the ground. This task was implemented in three different ways: (a) a homogeneous leaderless swarm operating within a convex hull of waypoint beacons, (b) a single leader swarm that was lead to the theater of operations by a sequence of waypoint beacons and then detected areas of water near the target beacon and (c) a repeat of task (a) using attractive swarm agent beacons to mark locations with water.

In all of these tasks, the swarm agents were outfitted with a conductivity sensor that was integrated into the Easysen board and basically consisted of a number of conductors that were dragged behind the swarm agent. The impedance between the conductors was sensed in order to determine the presence of water.

Task (c) is pretty much the exact implementation of what was described in the simulations before (Fig. 7 shows four phases from a test run). Since the material to be detected did not drift, we used a set of waypoint beacons to define the operating space. The four way point beacons enclosed a polygon of about 10000sqft. The turn radius was chosen to be about 0.3- 0.5 meters. The repel beacon was detectable with a maximum range of about 2-4 meters depending on orientation. The swarm size was between 10 and 15 agents.

Task (b) consisted of leading a swarm along a linear arrangement of way points from a base to a target way point beacon.

This was accomplished with the help of a single leader and a swarm of 12 followers. In this scenario, the followers only recognized one attractive beacon, namely the leader beacon. Once the leader arrived at the destination, it turned the swarm of followers over to the target beacon (by passing the leader ID to the target way point) and removed itself from the scene. After the water detection task was accomplished, the leader moved back to the target beacon and took over again. It then led the followers back to the base using the inverted sequence of way points that was used to move to the target location previously.

4.2 Protection/Coverage of GroundArea and Interception of Hostile Agents

The task is to protect a target area with well defined dimensions from incoming aerial objects (e.g., hostile fighters, UAVs, rocket-propelled granates, missiles, etc.). The goal is to achieve a close to 100% intercept rate with a minimal number of protecting UAVs using the proposed beacon principle. Effectively, the UAVs will form a protective shield above the target area without having to be able to detect or track incoming objects. Whenever an incoming object attempts to penetrate the shield, at least one UAV will collide with it, thus destroying both, the UAV itself and the hostile agent on impact.

Fig. 8 (A) through (F) shows screenshots taken from various stages in the interception task run in our simulation environment. Friendly UAVs (green) have been deployed and are protecting a military installation (A). The green building in the center (equipped with an attractive beacon) is the critical target that needs to be protected from attacks by hostile agents. Note that the red center in friendly UAVs indicates that their tar beacons are on. (B) shows a close-up of a hostile agent (red) coming from north west in an attempt to penetrate the UAV shield. At any given time it is possible that multiple hostile agents are attacking the installation from multiple directions (C). (D) shows the interception of three hostile agents just before the collision, the fourth agent in the southwest avoids a collision (shown in (E)) and escapes (F).

After several short pilot simulations, in which we determined different parameter settings for the UAV system, we conducted 100 long-term simulation runs with different initial conditions of the above protection scenario, each of which lasted for 1 million update cycles (amounting to a total computation time of several days on a 44 node Beowulf cluster). 90 UAVs with fixed repulsion range p = 78 were randomly position over the target area and had 500 cycles to organize themselves, after which point hostile agents entered from random locations every 100 cycles (regardless of whether the previous ones had been destroyed or not) with the goal of destroying the target building in the center. Whenever a hostile agent was destroyed, a new UAV was dispatched (assuming that the hostile agent collided with an existing UAV). The speed of hostile agents was 30 units/cycle, whereas the speed of UAVs was 0.7 units/cycle (i.e., a ratio of 1:42).

The obtained results are very promising: (1) no friendly UAV collided with another UAV in any run; (2) no hostile agent was able to destroy the target in any run; (3) all 9995 hostile agents were destroyed; and (4) the number of alive UAVs at the end of each 1 million cycle run was higher than the original number (on average by 63) because sometimes two hostile agents collided at the same time with the same UAV.

Robotic implementation: This task has been implemented partially using the swarm agents described before and a single attractive waypoint beacon marking the high value target. In fact, we used a two-tier swarm with two different densities and demonstrated that the dense swarm agents (with the smaller repel radius) always position themselves closest to the attractive waypoint beacon, whereas the low density swarm agents operate on the perimeter (see Fig. 9). This emergent behavior occurs regardless of the initial conditions of the swarm and shows that the swarm can form a tight coverage of the area (with the high density close to the area that needs to be protected). Note that we have not demonstrated the intercepting capability of the robotic swarm (as performed in the simulations where swarm agents get destroyed) for obvious reasons.

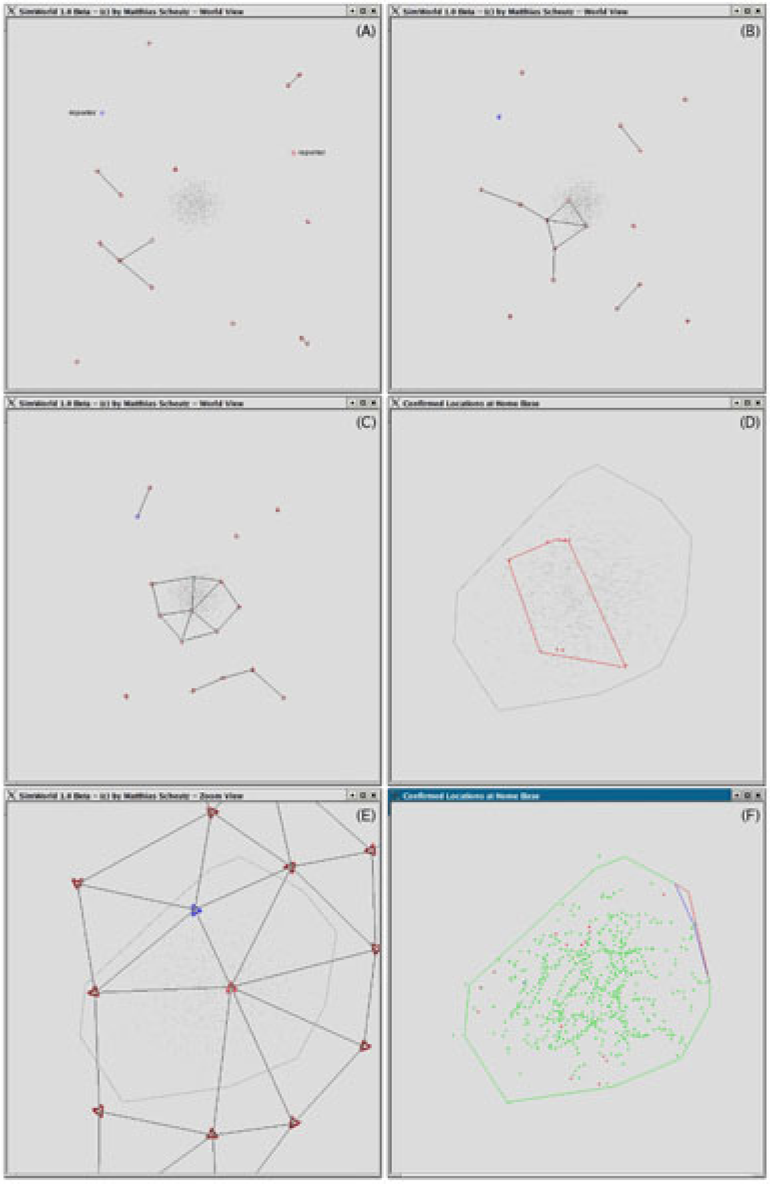

Figure 5: Different phases in the target sensing and location reporting task (see text for an explanation).

4.3 Mixed UGV-UAV Target Tracking, Target Identification, and Target Enclosure

All of the above mentioned UAV/UGV systems can be combined into mixed UGV-UAV system. For example, UAVs might be able to detect potential targets at a distance, but may not be able to identify them. UGVs, on the other, may be able to analyze and enclose targets once they are close-by, but cannot themselves find them. A mixed UGV-UAV system can utilize the strengths of each and form a powerful combined system that addresses (1) surveillance and reconnaissance (on the UAV part), (2) target detection and tracking (again UAVs), (3) target identification and enclosing (UGVs), (4) data collection and reporting of data back to the base station (4) (UGVs together with UAVs).

To illustrate how our proposed principle naturally extends to mixed UGV-UAV teams, consider the simulated scenario depicted in Fig. 10 (A) through (F), where unrecognized objects (four gray circles) are moving east in hostile territory. A (blue) UAV detects the four unidentified moving objects (A) and turns on its attractive beacon (red spot in the center). Shortly thereafter (B), four other UAVs that were attracted to the beacon signal, join the first UAV and form an adhoc wireless UAV network (indicated by blue lines connecting the UAV). The larger red UAV has long-distance wireless capacity and can report data to the home base. Note that all UAVs have turned their beacons on. (C) shows UGVs (green), which were dispatched in response to the information communicated by the red UAV to enclose the moving target and are now attracted to the target area by the UAV beacons. The close-up in (D) shows how UGVs are forming a wireless UGV net work (indicated by green lines). Once within communication range, UGVs and UAVs form a mixed wireless adhoc network (indicated by black lines in addition to green and blue lines) (E). As soon as UGVs come within target recognition distance and are able to identify hostile targets, they turn on their beacons (red spots in the center of the UGVs) (F), which reinforce their positions. As part of the tracking and identification, UGVs have enclosed the hostile agents, are collecting data and use the mixed adhoc wireless network to report the data to the home base (via the red UAV).

4.4 Evidential Reasoning and Filtering

So far, navigation for resource allocation and collision avoidance was solely based on radio beacon signals. The same principle can be applied to other signal modalities or even mixes of different signal modalities without a significant increase in system complexity. Examples would be IR radiation, sound, light, nuclear radiation, etc. Many applications require even better discrimination of target activity and the incorporation of a signature detection strategy can lead to further increases in performance. This of course requires schemes that are capable of detecting, and discriminating among, target signatures that may be buried in 'noise' or 'routine' activity (e.g., detection of potential threat situations). Attaining success in this endeavor with an acceptable level of veracity is an extremely challenging task.

To increase the sensitivity to such low-signature events, it is imperative that all available and pertinent information be exploited to the fullest extent possible. Source diversity, viz., the ability to handle different sensor/source modalities, enables the system to 'cross-correlate' information from a large number of data sources, thus increasing its ability to detect a low signature event that would otherwise have gone unnoticed. But, there are other dimensions of information diversity that can be exploited: temporal diversity and spatial diversity. The time at which an event was recorded can give very valuable information, especially when it is recorded over a long time horizon. In a similar manner, spatial diversity can also significantly enhance the sensitivity of a system to low signature events.

Traditional temporal, spatial and spatio-temporal filters operate within the "signal space" before raw signals are fused. Signature detection at a given location for instance is carried out by "pooling" signals from sensors of the same modality. Data imperfections (e.g., a compromised sensor, missing data and other data ambiguities) must then be accommodated after the signature detection phase.

Evidence filtering is a radically new strategy that operates within the "evidence space" generated after fusion [38]. It is developed in the Dempster-Shafer (DS) belief theoretic framework, which has been utilized for uncertainty modeling in numerous applications over the years [41, 42]. DS theory is particularly well-suited for the detection of low signature events (e.g., potential threats) because of its ability to quantify potentially critical, more qualitative aspects of evidence - probabilistic approaches, which require one to make initial assumptions (e.g., independence of events, initial distributions, etc.), do not appear to be well-suited for combinations of temporally sequenced evidence as is needed for a swarm system.

Evidence filtering thus enables collecting, processing and mining evidence for low-signature events that may have been generated from possibly a large number of sources

* having different degrees of reliability;

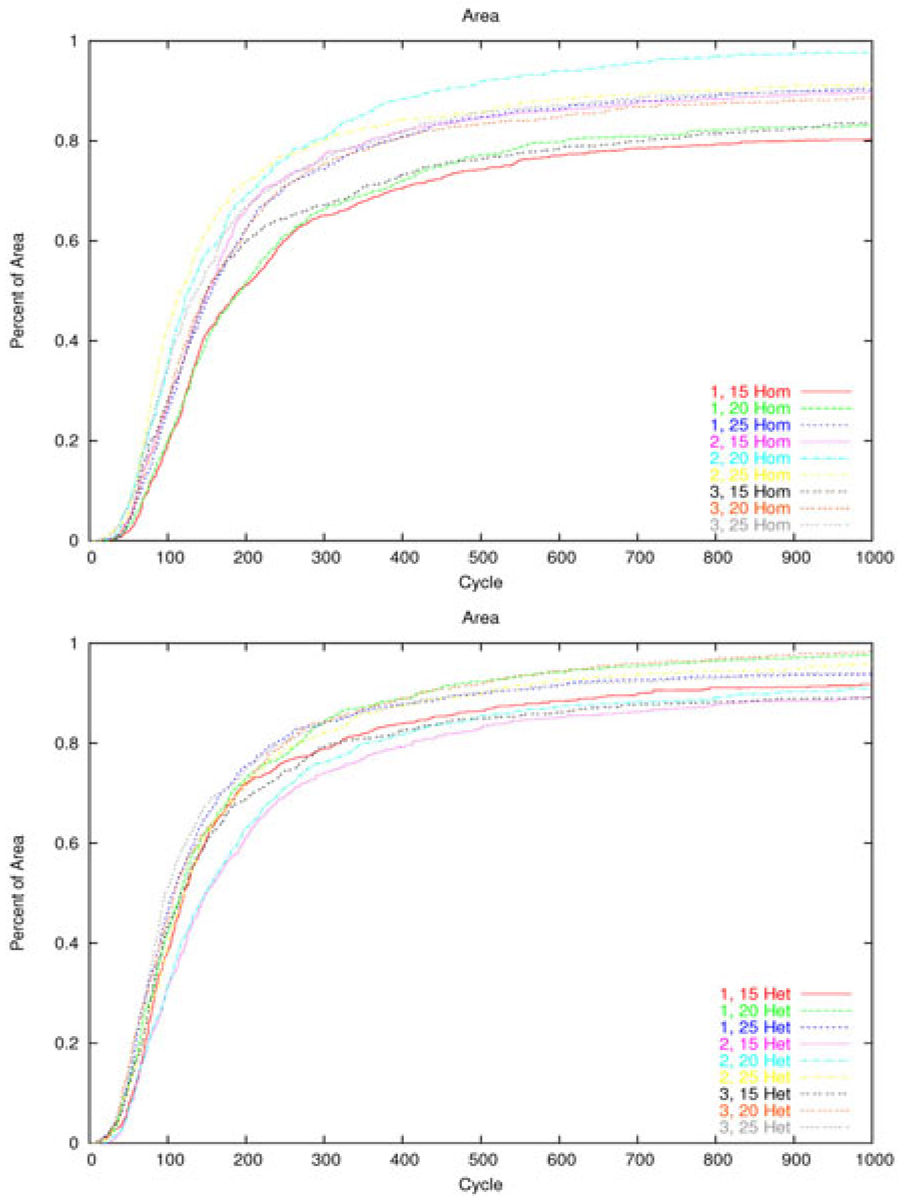

Figure 6: The percentage of the actual target area reported by the sensor network as function of simulation cycles for homogeneous (top) and heterogeneous (bottom) units (1=100%).

* characterized by a significant qualitative component;

* generating possibly streaming data; and

* possessing various data imperfections (e.g., ambiguous and missing data, compromised sensors, data that are not temporally synchronized, etc.).

With the recently developed evidence strategy that can account for source heterogeneity [39, 40], we can also accommodate sources possessing vastly different 'scopes of expertise' (e.g., heterogeneous agents, etc.).

In evidential filtering, information modality, space and time are all exploited to produce a fusion process that is highly selective enabling given event signatures to be searched in real-time; data imperfections are better accommodated from the very outset rendering the processes of evidence mining and knowledge discovery highly robust. Since they operate on evidence that is generated from all the available modalities, the information base that evidential filters have access to is significantly larger than if it had access to information generated from only one modality. This we believe is the key for its superior performance. Fig. 13 compares and contrasts the evidence filter with the traditional filter. Signature Triggered Mobile Sensor Beacons: The proposed navigation and agent coordination scheme provides significantly improved target detection capability through the use of radio beacons that are triggered only if certain signal signatures in space, time, or modality are present. Target detection capability is further improved by using a sensor fusion scheme that is based on "evidence filters", i.e. signature specific, tuned filters that selectively fuse sensor information based on DS Theory.(However, other sensor fusion schemes could also be plugged into this algorithm.) In addition to improved target detection, this principle also allows to specify more complex target characterizations, while still keeping the supporting hardware requirements very low. (In fact, a simple firs update scheme in space, time, or both can easily be implemented using analog hardware, while more complicated spatio-temporal schemes usually require a digital implementation.) This enables the sensor swarm to perform highly efficient resource allocation using the same fundamental navigation principles as in task 1. The only difference between the basic navigation system in task 1 and the signature triggered one in task 2 is the condition for triggering a beacon.(In fact the signature detection algorithm can be considered to be a "plug-in" for the navigation principle in task 1.) This signature triggered beacon principle is important in situations where an area of interest is described by sets or classes of "interesting signal signatures in space or time". In many cases, these signatures cannot be characterized well by presence/ absence of a certain signal modality or even signal modality mixes. What is exploited in this concept is the spatio-temporal correlation of signals across several modalities.

In order to illustrate the principle, consider Figure 11, which shows a small and well defined theater of operations with areas that exhibit certain properties that can be sensed if the sensors are sufficiently close. (A modified version of the same principle can be applied to the case where the signal source cannot be sensed from even a small distance). In order to explain the triggering algorithm for a beacon, it is sufficient to consider the single UAV case. There are four different areas in Figure 11 that exhibit four different modality mixes which can be detected if the UAV is within the marked area. However, the quality of the detection results vary and deteriorate from the center towards the boundaries of these areas. In other words, the red area stands for detectable modality mix "red", the green area stands for detectable modality mix "green", etc. and detection deep inside these regions is better than on the periphery. In order to illustrate the idea, we use the following simple signature characterization: Search for the presence of four different modality mixes (from a total of possibly many modality mixes) that are in close proximity of each other. Once four different mixes are found to be present with a sufficient degree of certainty quantified by the belief measures, activate the beacon. (In this simple example, temporal dependency is not exploited and spatial correlations are not further specified except to say that the four different mixes have to occur in four regions that are in close proximity. More complex spatiotemporally dependent signature can be specified in a similar manner.)

Figures 11 and 12 shows the initial random walk of the UAV and the sequence of detection the modality mixes "red", "black", "purple", and finally "green". What is shown in Figures 11 and 12 are different stages of the same UAV trajectory. Note that the UAV only moves from one modality mix to the next if it has found sufficient evidence of the modality mix current being examined. How much evidence in the presence of "red" has been detected is quantified by the belief in red. The belief measures in "red", "black", "purple" and "green" over time are displayed in the time traces below the respective UAV trajectory portions. Note that in this example a threshold of 0.5 was needed in order to proceed form the present modality mix area to the next. After the last modality mix was detected with belief 0.5 the beacon is activated. A number of subsequent actions can now be taken (not shown in Figures 11 and 12):

(a) The UAV can activate its beacon (for a preset amount of time) to attract other UAVs that have an identical mission. After its own beacon is de-activated, the UAV goes back to searching for the same signature as it did at the very beginning. This way, a portion of all UAVs beacons are always "on" and tracking of the four different modality mixes can be achieved. (This is a generalization of the task 1 example.)

(b) The UAV can activate a beacon that is unique for the four different sets of modality mixes that were found, thus attracting only those UAVs that have "high sensitivity/high resolution" sensing capabilities for the detected signals.

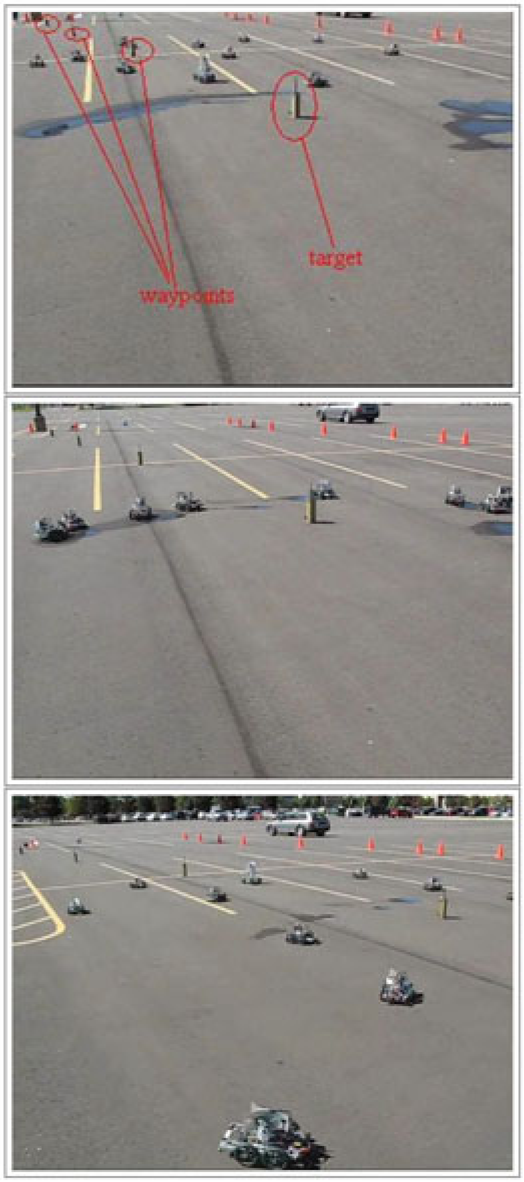

Figure 7: Three phases in the target detection and tracking task: the team leader leads the swarm to the target area (top); swarm agents successfully demarcate wet areas (middle); the leader re-activates agents to leave the area (bottom).

(c) The UAV could further explore spatio/temporal correlations of the detected signal cluster using spatiotemporal evidence filters.

There are many other options that one could employ in addition to the ones stated in (a)-(c).

5. RELATED WORK

Much research in distributed control and robotics has focused on controlling a large groups of autonomous agents. The different control schemes can be roughly put into three categories: centralized control, decentralized control, and local control.

Centralized control is well-known to have a number of problems, such as low fault-tolerance of the system (due to its dependency on many communication links), poor scaling (due to a common sink and command node), and high complexity (due to powerful processing centers), some of which are solved by distributed control (e.g., [23, 24, 25, 26, 27, 28, 29, 30, 31, 32,33, 34, 35]). Both, centralized and distributed navigation control schemes, require digital communication (possibly causing delays, congestion, packet drop, etc.) and often resort to GPS technology for trajectory control. While both paradigms are able to provide performance guarantees for individual agents (by controlling them separately and providing allocations of specific resources to specific targets), this kind of microscopic control of the entire system is often not needed (e.g., it does not matter which UAV intercepts a hostile agent).

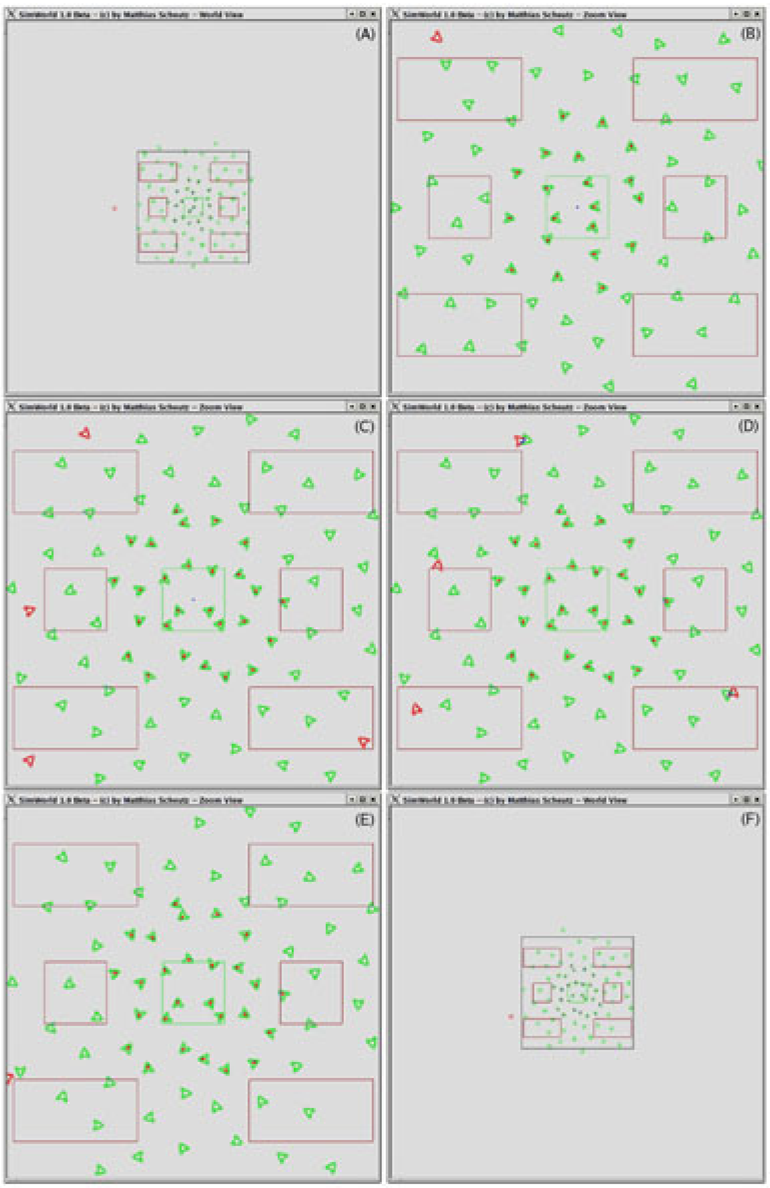

Figure 8: Different phases in the area coverage and interception task (see text for explanation).

Local control takes advantage of this fact, leading to emergent behavior that share many features of general distributed control, but do not suffer from scaling problems (only local neighbors are needed for communication) (e.g., [1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14]). While most local approaches have low complexity and good fault-tolerance, they still suffer from the consequences of using digital communications (e.g., [22]).

Our proposed approach (see also [1]) is at the ultra-low complexity end of the entire spectrum of local control (see the comparison in Table 1), leading to very low cost implementations that were previously not feasible: neither digital communication nor GPS are required for navigation, and decisions are based on simple reactive principles that can be implemented in analog hardware and with practically no delay. Of all autonomous swarm systems, the proposed principle, to our knowledge, offers the lowest cost and complexity, yet at the same time it can successfully accomplish a great variety of tasks and missions.

6. CONCLUSION

In this paper, we proposed an ultralow complexity versatile resource allocation and navigation principle for robotic swarms that is extremely simple and only involves local beacon-based interactions.

The principle scales since there is no digital communication involved for navigation (even though digital communication is used for gathering and pre-processing information). The underlying beacon-based mechanisms make the entire system extremely robust to individual agent failure, and the performance of the whole system degrades gracefully with a decreasing number of agents. Moreover, the principle provides simple and coarse performance prediction of the entire swarm in a stochastic sense (rather than each individual agent) even in cases of heterogeneous networks with different types of agents. Future work will develop a mathematical framework for obtaining formal results about the system properties.

Compared to other proposed solutions, the simplicity of the proposed system is its main strength. Because the principle is about as simple as it can get without losing important properties (e.g., collision-avoidance or control of swarm density), it can be implemented in possibly very small robots and it can be used to build large robot swarms, where each agent is cheap and expendable. We demonstrated both in simulations, and more importantly, in robotic implementations, the viability of the proposed control mechanisms. Moreover, we showed that the principle can be adapted to many different types of real-world swarm applications, including many application-specific systems for intelligence, surveillance, reconnaissance, target tracking and interception, which can be achieved by simply changing the beacon detection radius and/or making certain agents stationary (e.g., mobile sensor networks, multiagent target detection and tracking, patrolling for fixed surveillance tasks, interception tasks for infra-structure protection, targeting, plume tracking, ad-hoc cellular communication service facilitation, etc.). Future work will continue to investigate possible application areas of the principle, obtaining more detailed performance measures for specific applications (e.g., see [1,46,47,48,49,50] for a start).

7. REFERENCES

*References

[1] M. Scheutz and P. Schermerhorn and P. Bauer "The utility of heterogeneous swarms of simple UAVs with limited sensory capacity in detection and tracking tasks" Proceedings of IEEE Swarm Intelligence Symposium, pp. 257- 264, 2005.

[2] J.A. Sauter and R. Matthews and H.V.D. Parunak and S.A. Brueckner "Performance of Digital Pheromones for Swarming Vehicle Control" Proceedings of Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2005.

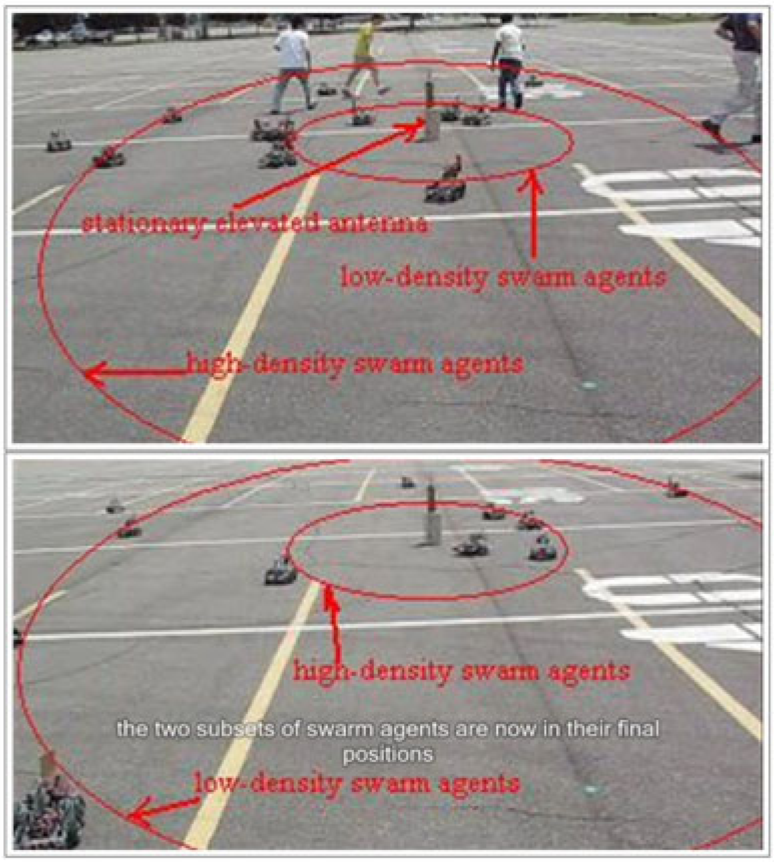

Figure 9: Swarm agents starting out with "high density agents" (i.e., agents with small ρ) on the outside and "low-density agents" (i.e., agents with a larger ρ) close to the beacon (top). After a while, the high-density agents gather around the beacon area while the low-density have moved to the outside (bottom).

[3] E. Bonabeau and M. Dorigo and G. Theraulaz Swarm Intelligence: From Natural to Artificial Systems Swarm Intelligence Oxford University Press, 1999.

[4] H.V.D. Parunak and S. Brueck ner. "Swarming Coordination of Multiple UAVs for Collaborative Sensing" Proceedings of Second AIAA Unmanned Unlimited Systems, Technologies, and Operations Conference, AIAA, 2003.

[5] T. Balch and R.C. Arkin. " Behavior based formation control for multi robot teams", IEEE Trans. on Robot. Autom., Vol. 14, pp. 926939, Dec. 1998.

[6] C.W. Reynolds. "Flocks, herds, and schools: a distributed behavioral model", Computer Graphics ACM SIGGRAPH 87 Conference Proceedings, 21(4):25-34, July 1987.

[7] V. Trianni and M. Dorigo "Emergent collective decisions in a swarm of robots" Proceedings of IEEE Swarm Intelligence Symposium, pp. 241- 248, 2005.

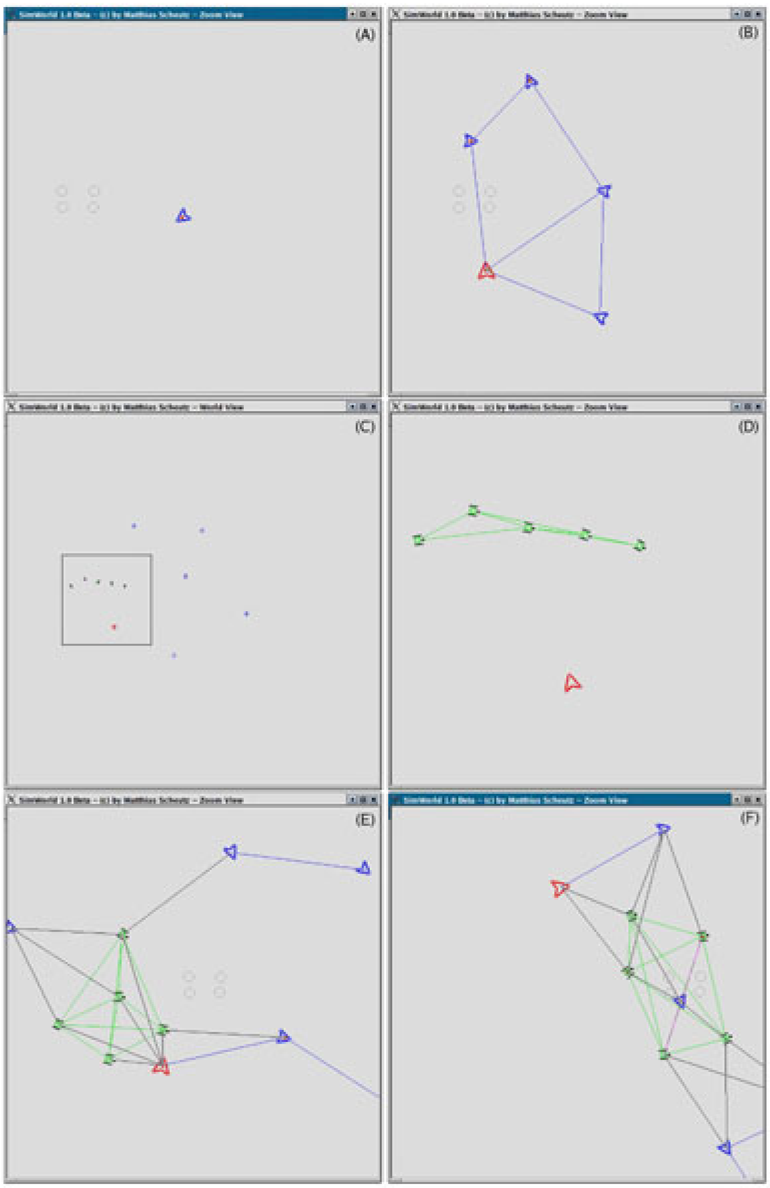

Figure 10: Different phases in the mixed UGV-UAV target detection, tracking, and enclosure task (see text for an explanation).

[8] D. Zarzhitsky and D.F. Spears and W.M. Spears, W.M. "Swarms for chemical plume tracing" Proceedings of IEEE Swarm Intelligence Symposium, pp. 249- 256, 2005.

[9] R.W. Deming and L.I. Perlovsky "Sensor fusion for swarms of small unmanned aerial vehicles" Proceedings of IEEE Swarm Intelligence Symposium, pp. 302- 308, 2005.

[10] "Evolving behaviors for a swarm of unmanned air vehicles" Proceedings of IEEE Swarm Intelligence Symposium, pp. 317- 324, 2005.

[11] A. Jadbabaie, J. Lin and A.S. Morse. "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules", IEEE Trans. on Automatic Control, Vol. 48, No. 6, 988-1002, 2003.

[12] J. Fredslund and M.J. Mataric. "A General, Local Algorithm for Robot Formations", IEEE Transactions on Robotics and Automation, Special Issue on Advances in Multi-Robot Systems, 18(5), pp. 837-846, Oct 2002.

[13] A. Howard, M.J. Mataric, and G.S. Sukhatme. "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks", Autonomous Robots, Special Ossue on Intelligent Embedded Systems, G. Sukhatme, ed., pp. 113-126, 2002.

[14] J.H. Reif and H. Wang. "Social potential fields: A distributed behavioral control for autonomous robots", Robot. Aut. Syst., Vol. 27, pp. 171-194, 1999.

[15] Y. Liu and K.M. Passino "Stability analysis of swarms in a noisy environment" Proceedings of IEEE Conference on Decision and Control, 4, pp. 3573-3578, 2003.

[16] H.G. Tanner. "Flocking with Obstacle Avoidance in Switching Networks of Interconnected Vehicles", IEEE International Conference on Robotics and Automation, New Orleans LA, April 26-May 1, pp. 3006-3011, 2004. Gaudiano, P.; Bonabeau, E.; Shargel, B.

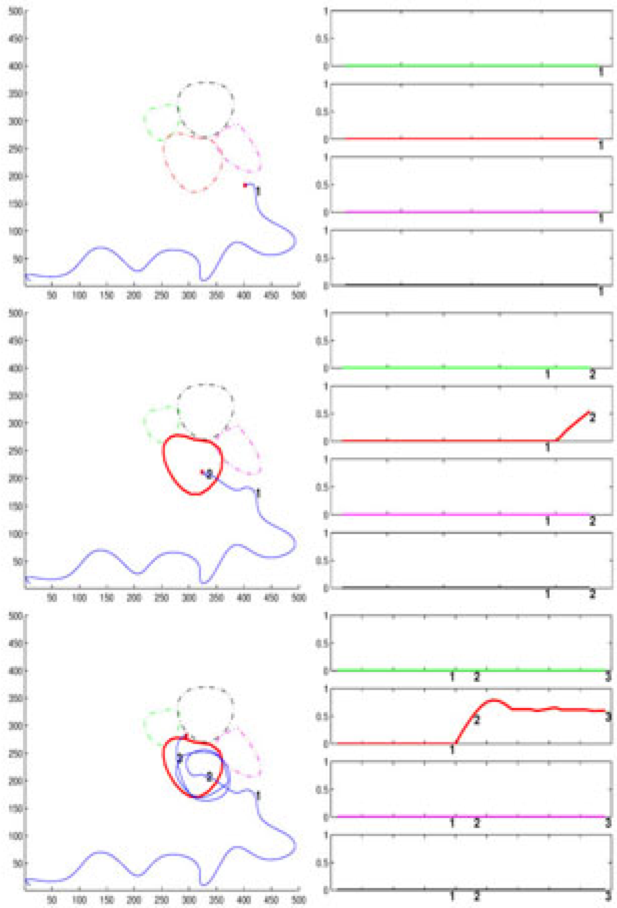

Figure 11: UAV trajectories at various stages of the evidence filtering process: paths (left) and filter activity (right), see text for details.

[17] H.G. Tanner, A. Jadbabaie and G.J. Pappas. "Stable Flocking of Mobile Agents, Part I: Fixed Topology", 42nd IEEE Conference on Decision and Control, Maui Hawaii, pp. 2010-2015, December 2003.

[18] Y. Liu, K.M. Passino and M. Polycarpou. "Stability Analysis of 1-D Asynchronous Swarms", Proc. of American Control Conference, 716721 (2001)

[19] Y. Liu and K. Passino. "Stable social foraging swarms in a noisy environment", IEEE Trans. on Autom. control, Vol. 49, No. 1, pp. 30-44, January 2004.

[20] V. Gazi and K. Passino "Stability analysis of swarms", IEEE Trans. on Autom. Control, Vol 48, pp. 692697, April 2003.

[21] V. Gazi and K. Passino. "A class of attraction/repulsion functions for stable swarm aggregations", CDC 2002, Las Vegas, pp. 2842-2847, 2002.

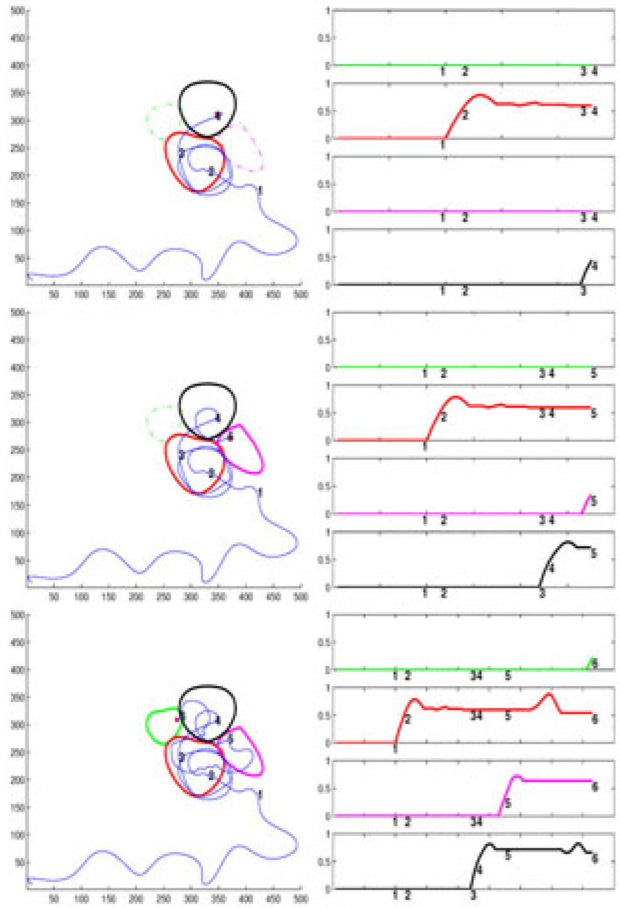

Figure 12: UAV trajectories at various stages of the evidence filtering process: paths (left) and filter activity (right), see text for details.

[22] D. Garagic "On delay-dependent stability of a swarm of networked autonomous vehicles under communication constraints" Proceedings ofIEEE SwarmIntelligence Symposium, pp. 289-293, 2005.

[23] Y. Yang and A.A. Minai and M.M. Polycarpou "Decentralized Cooperative Search in UAVs Using Opportunistic Learning" AIAAGNC Conf. 2002, pp. 2002-4590, 2002.

[24] W.M. McEneaney and B. Fitzpatrick "Control of UAV Operations under Imperfect Information" Proc. ofAIAA Technical Conf. and Workshop on Unmanned Aerospace Vehicles, Systems and Technology, pp. 2002-3418, 2002.

[25] M. Flint and M. Polycarpou and E. Fernandez-Gaucherand "Cooperative Path-planning for Autonomous Vehicles using Dynamic Programming" 15th Trienal World Congress, IFAC, 2002.

[26] M. Flint and M. Polycarpou and E. Fernandez-Gaucherand "Cooperative Control for Multiple Autonomous UAVs Searching for Targets" Proceedings of 41st IEEE Conference on Decision and Control, 1, 2002.

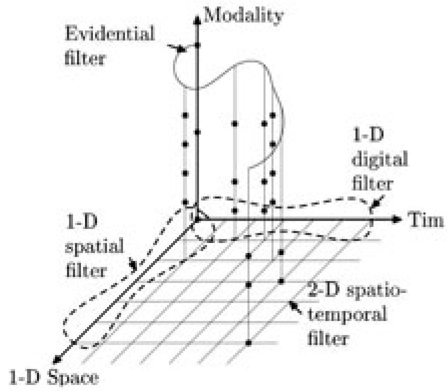

Figure 13: Unlike traditional filters, evidence filters can exploit all available modalities.

[27] M. Pachter and J.J. D'Azzo and A.W. Proud "Tight Formation Flight Control" Journal of Guidance, Control and Dynamics, 24, No. 2, pp. 246-254, 2001.

[28] E. Rimon and D.E. Koditschek "Exact Robot Navigation Using Artificial Potential Functions" IEEE Trans. on Robotics and Automation, 8, 5, pp. 501-518, 1992.

[29] M. Flint and M. Polycarpou and E. Fernandez-Gaucherand "Cooperative control for multiple autonomous UAV's searching for targets" Proceedings of 42nd IEEE Conference on Decision and Control, 3, 2823-2828, 2002.

[30] L. Jiecai "Some new optimal control problems in UAV cooperative control with information flow constraints" Proceedings of the American Control Conference, 3, pp. 2181-2186, 2003.

[31] R.W. Beard and T.W. McLain "Multiple UAV cooperative search under collision avoidance and limited range communication constraints" Proceedings of 42nd IEEE Conference on Decision and Control, 1, pp. 25-30, 2003.

[32] P.R. Chandler and M. Pachter and D. Swaroop and J.M. Fowler and J.K. Howlett and S. Rasmussen and C. Schumacher and K. Nygard "Complexity in UAV cooperative control" Proceedings of 42nd IEEE Conference on Decision and Control, 3, pp. 1831-1836, 2002.

[33] J.S. Bellingham and M. Tillerson and M. Alighanbari and J.P. How "Cooperative path planning for multiple UAVs in dynamic and uncertain environments" Proceedings of 42nd IEEE Conference on Decision and Control, 3, pp. 2816-2822, 2002.

[34] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. Hedrick. "An overview of emerging results in cooperative UAV control", CDC 2004, Nassau, Bahamas, 2004.

[35] D. Shim, H.J. Kim and S. Sastry. " Decentralized nonlinear model predictive control of multiple flying robots" CDC 2003, pp. 3621-3626, Maui, Hawaii, 2003.

[36] M. Scheutz and P. Schermerhorn. "Adaptive Algorithms for the Dynamic Distribution and Parallel Execution of Agent-Based Models". Journal of Parallel and Distributed Computing, 2006

	Navigation Control Schemes			
Properties	Centralized	Distributed	Local	Proposed
Complexity	high	med.	low	low
Indiv. Perf. Guarantee	high	high	low	low
Scalability	low	med/high	high	high
Adaptability	low	med/high	med	high
System fault-tolerance	low	med.	med/high	high
Model-independent perf.	low	med.	med/high	high
Computing requirements	high	med/high	med/low	ultra-low
Digital comm. bandwidth	high	high	med/low	none

Table 1: Comparison of properties of different control strategies.

[37] SWAGES. http://www.nd.edu/~airolab/software/

[38] D.A. Dewasurendra, P.H. Bauer and K. Premaratne, "Distributed Evidence Filtering in Networked Embedded Systems", Proceedings of the Workshop on Networked Embedded Sensing and Control (NESC'05), 2005.

[39] E.C. Kulasekere, K. Premaratne, D.A. Dewasurendra, M.-L. Shyu and P.H. Bauer, "Conditioning and Updating Evidence", International Journal of Approximate Reasoning, 36(1), 75-108,2004.

[40] K. Premaratne, D.A. Dewasurendra and P.H. Bauer, "Evidence Combination in an Environment with Het erogeneous Sensor", IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, to appear in 2006.

[41] P. Smets, "Practical uses of belief functions", Proceedings ofConference on Uncertainty in Artificial Intelligence, 621-621, 1999.

[42] D.L. Hall and J. Llinas, eds. Handbook of Multisensor Data Fusion. Boca Raton, FL: CRC Press, 2001. "Practical uses of belief functions", Proceedings of Conference on Uncertainty in Artificial Intelligence, 621-621, 1999.

[43] M.A. Kovacina and D. Palmer and G. Yang and R. Vaidyanathan "Multi-agent control algorithms for chemical cloud detection and mapping using unmanned air vehicles" IEEE/RSJ International Conference on Intelligent Robots and System, 3, pp. 2782-2788, 2002.

[44] R.W. Beard and T.W. McLain and M. Goodrich "Coordinated target assignment and intercept for unmanned air vehicles" Proceedings of IEEE International Conference on Robotics and Automation, 3, pp. 2581-2586, 2002.

[45] C.A. Lua and K. Altenburg and K.E. Nygard "Synchronized multipoint attack by autonomous reactive vehicles with simple local communication" Proceedgins of IEEE Swarm Intelligence Symposium, pp. 95-102, 2003.

[46] R. Connaughton and P. Schermerhorn and M. Scheutz "Physical Parameter Optimization in Swarms of Ultra-Low Complexity Agents" Proceedings of Autonomous Agents and Multi-Agent Systems AAMAS 2008, pp. 1631-1634, 2008

[47] D.A. Dewasurendra and P. Bauer and M. Scheutz and K. Premaratne "Evidence Based Navigation in Swarms" Proceedings of CDC 2006, pp. 5078-5083, 2006.

[48] C. Middendorff and M. Scheutz "Real-time Evolving Swarms for Rapid Pattern Detection and Tracking" Proceedings of Artifical Life X, pp. 419-426, 2006

[49] M. Scheutz and P. Bauer "A Scalable, Robust, Ultra-Low Complexity Agent Swarm for Area Coverage and Interception Tasks" Proceedings ofISIC06, pp. 1258-1263, 2006.

[50] M. Scheutz "Real-time Hierarchical Swarms for Rapid Adaptive Multi-Level Pattern Detection and Tracking" Proceedings of the 2007 IEEE Swarm Intelligence Symposium, Honolulu, Hawaii, 234-241, 2007.

Matthias Scheutz

Tufts University Medford, MA 021555, USA

Tel.: +1-617-627-0453

Fax: +1-617-627-2227

matthias.scheutz@tufts.edu

Peter Bauer

University of Notre Dame Notre Dame, IN 46556, USA

Tel.: +1-574-631-8015

pbauer@nd.edu

(1) Zigbee is a local-area wireless network designed to replace the increasing number of unique remote controls in consumer devices. Zigbee-based products operate in three unlicensed bands worldwide, including 2.4 GHz (global), 902 to 928 MHz (Americas), and 868MHz (Europe). Data rates of 250 kbps can be achieved at 2.4 GHz (16 channels), 40 kbps at 915 MHz (10 channels), and 20 kbps at 868 MHz (1 channel), achieving a transmission distance of well over 100 m. Channel width is 2 MHz with 5 MHz channel spacing.

(2) Depending on the specific configuration, establishing some of these properties will be challenging and require new proof techniques (e.g., UAVs might fly at a constant minimal speed, hence many classical techniques for collision-free navigation, especially those based on potential fields, are either not directly applicable or difficult to apply [15, 16, 17, 18, 19, 20, 21]).

Table 1: Comparison of properties of different control strategies.

Navigation Control Schemes

Properties	Centralized	Distributed	Local	Proposed
Complexity	high	med.	low	low
Indiv. Perf. Guarantee	high	high	low	low
Scalability	low	med/high	high	high
Adaptability	low	med/high	med	high
System fault-tolerance	low	med.	med/high	high
Model-independent perf.	low	med.	med/high	high
Computing requirements	high	med/high	med/low	ultra-low
Digital comm. bandwidth	high	high	med/low	none

Source Citation (MLA 9th Edition)

Scheutz, Matthias, and Peter Bauer. "Ultra-low complexity control mechanisms for sensor networks and robotic swarms." *International Journal of New Computer Architectures and Their Applications*, vol. 3, no. 1, Jan. 2013, pp. 86+. *Gale Academic OneFile*, link.gale.com/apps/doc/A353439973/AONE?u=googlescholar&sid=bookmark-AONE&xid=fe7ed7f0. Accessed 6 Feb. 2023.

Gale Document Number: GALE|A353439973