CONTROL OF UAV SWARMS: WHAT THE BUGS CAN TEACH US

Paolo Gaudiano, Benjamin Shargel, Eric Bonabeau
Icosystem Corporation
Cambridge, MA

and

Bruce T. Clough
Air Force Research Laboratory
Control Sciences Division
Wright-Patterson AFB, OH

ABSTRACT

This article describes research in the control of UAVs. Control strategies exist for individual UAVs, but not for large, coordinated teams. While there is a general agreement that a cooperative, team or swarm approach to UAV control can be beneficial for a number of military and civilian applications, little work has been done to understand if and when it even makes sense to use more than a single UAV, or a few, individually controlled UAVs. We have begun a study of UAV control strategies that are based on swarm intelligence, a methodology that draws its inspiration from the behavior of social insects. Our long-term goal is to gain a systematic understanding of distributed control strategies, and a quantitative methodology to evaluate performance of UAV swarms under a variety of conditions. In this paper we outline our approach, describe an agent-based model we have used to develop and test our methodology, and present some of our results to date.

INTRODUCTION

The Air Force is investigating the development of teams of unmanned air vehicles (UAVs) to carry out search, suppression and other missions in high danger scenarios that could threaten the safety of military personnel. Current techniques for multi-UAV control, which rely on centralized control and on the availability of global information, suffer from severe limitations owing to the extreme complexity that arises from the interactions between swarm elements. Traditional, centralized approaches frequently lead to exponential increases in communication bandwidth requirements and in the size of the controlling software.

In contrast, swarms of simple biological or artificial organisms can exhibit rich emergent behaviors without the need for centralized control or global communication\(^1\). Swarms of living organisms often self-organize into highly complex systems: flocks of birds, schools of fish and swarms of insects offer clear examples of self-organized, emergent behaviors arising from the interaction of many simple individuals.

While the concept of swarming has received significant attention in both government and commercial settings\(^2,3\), several important questions must be solved in order for any swarm-based solution to be applicable to the control of UAVs, or to any other government or commercial problem. First, there is a common misunderstanding that a decentralized, swarm approach may imply a level of autonomy that is unacceptable for military missions. How can we hope to control the behavior of a swarm when, by definition, the control is decentralized? Second, while there is a general notion that swarms increase redundancy, robustness and flexibility, there has been no systematic evaluation of the performance benefits of using swarms. What missions are particularly well suited for swarms? How does performance depend on swarm size? Are there missions that could only (or should not) be solved with swarms? Third, there is no quantitative understanding of the practical limitations and trade-offs involved in using swarms. Does software complexity increase or decrease when using swarms? Do swarms require massive communications bandwidth? Is the eventual performance increase of using a swarm offset by concomitant increases in cost?

With the support of a Small Business Innovation Research (SBIR) contract from the Air Force Research Laboratory, we have begun a systematic investigation into these issues. Our long-term goal is to design decentralized control strategies for UAV swarms in such a way that a single operator can issue commands to the swarm as a whole, and these will automatically result in rules for the control of individual UAVs so that the swarm as a whole completes the mission.
The remainder of this paper is organized as follows. The next two sections describe the model we have developed, and the missions we wanted the UAVs to perform. Next, we describe our control strategies. We then present our results, and close the paper with a discussion.

MODEL OVERVIEW

In order to study UAV control strategies, we have developed an agent-based simulation tool. Our initial simulator allows simple but effective 2D and 3D visualization of UAVs and targets. The following is a list of the assumptions and functionality we designed into the simulator:

- The terrain is defined as a rectangular region (actually a parallelepiped in 3D), which for convenience is subdivided into a grid of arbitrary coarseness. The grid is used primarily to determine coverage and to track “pheromone” signals (see below) left by each UAV as it flies over the terrain.
- Each UAV is able to fly at variable speed (within adjustable bounds), with independent pitch and yaw control. Control dynamics are simplified by specifying a maximum turn rate, and the ability to increase or decrease thrust.
- Each UAV is equipped with various sensors:
  - One forward-looking, cone-shaped ground sensor with adjustable radius and angular aperture, to detect terrain and possible targets.
  - One circular sensor to detect the presence of other UAVs within a prescribed (adjustable) radius.
  - GPS-like positioning capability.
  - “Pheromone” sensor: each UAV can detect, within a small rectangular region centered on itself, how much each terrain cell has been covered by itself or by other UAVs (see later description).
- Each UAV is aware of the terrain boundaries and will turn as it approaches each boundary to remain within the target area.
- Global communication between UAVs is possible. Our simulations test different strategies, some of which rely on global communication and others do not.
- Targets can be static for testing area-coverage missions, or they can move for suppression missions.

The simulator is written in the Java language for portability. Through a mixture of command-line parameters, GUI widgets and built-in variables, it is possible to modify nearly every aspect of the terrain, UAVs and targets.

![Figure 1: Screen capture of the simulator. See text.](image-url)
The simulator is divided into three areas. The top-left area includes various widgets to control certain aspects of the simulation in real time, such as pausing and restarting, shuffling targets randomly, or modifying the dynamics of the UAVs. This area also displays the time elapsed, percentage of terrain covered, and percentage of targets identified. The bottom area is a 3D view showing the boundaries of the terrain being searched (black wireframe box), the UAVs (circles), the area swept by the UAV ground/target sensors (yellow triangles), and the targets (red/green squares). Each UAV has a red vertical line connecting it to the ground to help visualize its position, a black line indicating its current heading, and a yellow line indicating its “desired heading.” The color of the UAVs is indicative of their current state, as described in a later section.

Finally, the grid in the upper right-hand corner is a top-down matrix representation of the terrain, which shows the grid used to determine coverage, the x,y position of the UAVs (black squares) and targets (red if not found, green if found), and a blue trace of varying intensity that represents the “pheromone,” i.e., the degree to which a given cell has been flown over by UAVs.

During a typical run, the user invokes the simulator with a number of run-time flags to modify the default simulator parameters, such as: terrain size, number of UAVs and targets, duration of the simulation, and several parameters related to the strategy used by the UAVs. The simulation can also be run without the GUI for faster execution. This is useful, for instance, when running multiple scenarios or even the same scenarios multiple times with varying random seeds to obtain statistically meaningful results.

**MISSION OVERVIEW**

We used the model to test UAV control strategies for two different missions: a search or area-coverage mission, in which targets are stationary and randomly distributed throughout the target area; a suppression mission, in which the UAVs have to strike targets that are moving at random speed and in random directions. In both cases, all UAVs are initially launched from a single point along the perimeter of the search area.

In a search mission, each UAV flies over the target area, sweeping its ground sensor over the terrain. Because the targets are stationary, the number of targets detected is directly proportional to the amount of area covered by the UAVs. The measure of performance is the percentage of area covered, or the percentage of targets identified.

In the suppression mission, UAVs begin by searching for targets. When a UAV finds a target, it switches to tracking mode, and it broadcasts a request for support: before attempting to strike a target, it must wait for another UAV to join it near the target, so that Bomb Damage Assessment can be performed once the UAV dives into the target. The UAV in the simulator turns yellow to indicate it is in tracking mode. Whichever UAV first responds to a request for support will fly toward the requestor. In the simulation tool, supporting UAVs turn magenta (not seen in the figure above). Once the supporting UAV is sufficiently close to the target, the requestor switches to strike mode (turns red), diving into the target. The strike is successful (target killed) with some probability, usually 80% in our simulations. If the target is killed, it turns green and the supporting UAV switches back to target search mode (black). If the strike fails, the supporting UAV now switches to active target tracking, and broadcasts a request for a new supporting UAV.

There are various performance measures we have used for suppression missions, such as: percent killed (PK), probability of killing the ith target (PKi), and time elapsed before killing ith target (TKi).

**SWARM CONTROL STRATEGIES**

One key question in understanding swarm control is the relative success and efficiency of various swarm (decentralized) strategies. By swarm or decentralized we mean a strategy in which each UAV independently receives some information and takes an action. In contrast, a centralized swarm control strategy might use an off-line optimization algorithm to define an explicit path for each UAV to follow.

Within the realm of decentralized control strategies, a further distinction that must be made is whether the UAV’s decisions are based on information that is collected in the UAV’s immediate vicinity, or potentially from the entire environment (see next subsection). We have devised several simple strategies based only on information available in the immediate surrounds of each UAV, and some strategies that took into account information gathered from the entire search area. We tested each strategy individually and some combinations of strategies.

- The baseline strategy is a condition in which one or more UAVs are flying in a straight line until they reach a boundary of the search area, at which time they turn to avoid
exiting the area. This strategy is used as a comparison for the other strategies.

- The random strategy is similar to the baseline, but at each time step each UAV can change its heading by a small random angle.
- In the repulsion strategy, each UAV can sense other UAVs within a given radius, and it maneuvers so as to keep other UAVs outside of that repulsion radius.
- The pheromone strategy assumes that, whenever a UAV flies over a terrain cell, it leaves a marker indicating that the cell has been visited. Other UAVs are then able to determine, within a small local area immediately around them, whether cells have been visited or not. The UAV can then make small adjustments to their flight pattern to favor flying over unexplored cells.
- In the global strategy, we assume that the search space is divided into a number of large, square sectors, and that a central controller monitors the level of coverage within each sector, as well as the number of UAVs currently in that sector or already heading there. UAVs constantly monitor the value of their own sector and compare it to the “value” of other sectors, which is a function of coverage, occupancy, and distance. If a more desirable sector is found, the UAV turns light blue and heads to the new sector.

Global vs. Local Information

Before reporting our results, it is important to clarify a distinction that is completely orthogonal to the local vs. global nature of the decentralized strategies we just listed: the kind of information that is available to a UAV can itself be local or global in nature. For example, a sensor sweeping the ground ahead to look for targets is returning local information. On the other hand, a number indicating the overall coverage rate for the entire search area represents global information. A pheromone marker of the type we proposed could in principle be obtained either locally or globally: if UAVs were capable of dropping and later detecting some sort of marker (a classical example of stigmergy), then the information would be truly local. However, such technology is not readily available, but the same result can be obtained if each UAV is equipped with GPS and with the ability to communicate to a central computer that tracks which cells have been visited. In either case, the control strategy is local, in that the UAV only considers pheromone information from a small area localized around itself.

RESULTS

Area Coverage

For all the area-coverage simulation results we used a search area of fixed size (2,000 units on each side and 1,000 units high, with a maximum UAV speed of 5 units of displacement per time unit) and we let each simulation run for 1,000 time units. At the end of the simulation we recorded the percentage of cells that had been visited during the simulation. A cell is considered to be visited if it has been detected by a UAV at least five times. The reason for requiring multiple detections is that each UAV can detect a given cell in its sensor field several times during a single pass.

We ran simulations to test each of the five strategies described above, and every possible combination of strategies. Here we provide a summary of the main results.

Baseline strategy

The results of the baseline case are shown in Figure 2. The left graph shows the percentage of the search area covered in 1,000 time units as a function of the number of UAVs. A single UAV covers only about 7.5% of the search area. As expected, larger swarms cover larger fractions of the search area, with 10 UAVs covering 44.8%. However, it is clear that the fraction of search area covered does not scale linearly with swarm size, as shown in the plot on the right. In fact, the larger swarms get less and less efficient in terms of coverage per UAV.

This result is not surprising, because as the number of UAVs increases, so does the probability that one UAV will fly over the tracks of another, which does not increase the overall coverage. However, it is useful to consider this as a way to measure the relative efficiency of a swarm compared to the efficiency of a single UAV.
Testing additional strategies to improve coverage

A simple modification of the baseline strategy is to add noise to the UAV motion (jitter), and to add some simple repulsion when UAVs come too close to one another. Through some systematic tests, we found that a jitter of +/-3deg/sec generates UAV movements that are still largely straight but with some variation that promotes a more thorough dispersion over the search space. We also found that creating a repelling force between UAVS closer than 30 units was most effective. However, the improvements afforded by these modifications were not significant except in particular conditions, such as with a small number of UAVs or a large search area.

The jitter and repulsion factors are hardly what one would consider a control strategy. Nonetheless, they clearly impact the efficiency of coverage. The next strategy we considered is loosely inspired on the concept of stigmergy, which is the term to describe indirect communication through the environment, such as the way in which ants can communicate with other ants by leaving pheromone trails on their way to and from food sources. As mentioned earlier, we assumed that UAVs mark the cells they have visited with a sort of pheromone, and that they can also detect the presence of pheromone in their vicinity.

The control strategy consists of finding uncovered cells within a small rectangular area centered around each UAV. The uncovered cells are added up to form a vector that attracts the UAV toward unexplored areas in its immediate vicinity.

We have found that the pheromone strategy is the crucial element to the superior performance shown here. Qualitatively, Figure 3 reinforces this point by showing the behavior of UAVs during a single run.

The top matrix is the coverage trace of a run with 10 UAVs and a 60-unit repulsion radius. The matrix on the bottom is also with 10 UAVs, but they use a local pheromone strategy and no repulsion. Clearly, the pheromone strategy pushes UAVs to search in a much more elaborate pattern that leaves less unexplored space.
When we combine jitter and repulsion with the pheromone strategy, the performance of the swarm improves significantly. Figure 4 shows the quantitative results obtained with this combination of strategies.

Efficiency of Swarm Strategies

Per-UAV Efficiency of Swarm Strategies

Figure 4: the use of a local “pheromone” strategy yields superior results.

One interesting point is worth mentioning here: it is evident from the figure above that the benefits brought on by the pheromone strategy are more pronounced as the swarm gets larger. This seems sensible, because effectively the UAVs are cooperating by sharing information through the environment. This type of improvement is indicative of a strategy that is meaningful in a swarm context.

It is also worth pointing out that we did not make a significant effort in trying to improve or optimize the pheromone strategy. For instance, adding up all the cells in a rectangle as we do can lead to occasional incorrect behavior, such as running around in tight circles when flying over a small cluster of unexplored cells, or going straight if there are equal numbers of unexplored cells to the left and to the right, even though the cells directly ahead might have been explored already. The incremental improvement of using a pheromone strategy at all was much greater than changes resulting from further modifications.

Scaling to larger UAV swarms

The simulation results described above were designed in part to test systematically the impact of UAV swarm size on the efficiency with which the task is carried out. We limited our swarm size to 10 so that meaningful comparisons could be made without having to change search area size, and in order to keep computational time within reasonable bounds. In one experiment we tested performance on the same task when the swarm size varied between 10 and 110 UAVs. The search area is increased to 6000x6000 units (to avoid saturation), and the UAVs make use of the pheromone strategy with repulsion. The results are shown in Figure 5.

Figure 5: Coverage as a function of swarm size for larger swarms.

Figure 5 shows that total coverage continues to increase with swarm size, although the relative efficiency decreases (note that the search area was significantly larger, so the exact numerical values should not be compared to those in previous figures).

Suppression Missions

We have extended the behavior of the UAV swarm to suppress mobile targets. Whereas in the area-coverage mission the UAVs were constantly in an exploration state, now they have a number of potential behavioral states and arbitrary stochastic transition rules for switching between them, which can be specified as input to the simulation. Following is a list of specific extensions of we made to the model in order to carry out the new type of mission:

- Targets are mobile, moving randomly and at a slower speed than the UAVs. Because their movement is random, it is possible for them to
escape the target area. Once a target has been killed it no longer can move.

- To handle the uncertainty of target location, UAV pheromone, whose presence implies the degree of coverage over a given area, now dissipates over time. Thus, an area that has been marked as covered at one time step might not be marked in the future, reflecting the fact that a target might have moved around during the intervening period

**UAV Behavior State Transitions**

As mentioned in an earlier section, the suppression mission requires UAVs to exhibit behavioral states beyond simple search. Specifically, UAVs can switch between simple target search behavior (modulated by the control strategies described earlier), and four additional behavioral states: Track, Support, Attack and BDA. The behavioral state of a UAV determines how it updates its velocity vector each time step.

Transitions between behavioral states are triggered by signals received by the UAVs. These are not necessarily communication signals, but rather abstractions of cues a UAV can receive from itself, its physical environment or other UAVs. As not all signals are relevant to all states, after the description of each signal is a list of states within which the signal can be received, enclosed in square brackets.

- **Found New Target**: The UAV has detected a target below that no other UAV has as its current target (Search, Support)
- **Found Existing Target**: The UAV has detected the current target of one or more other UAVs (Search, Support)
- **Lost Target**: The UAV’s current target has left the target area (Track, Assess Damage, Attack)
- **Recruited**: A communication has been received from another UAV requesting support in monitoring its current target; by “monitoring” we mean is a general term that applies to the Tracking, Damage Assessment and Attack states, in which a UAV has homed in on a specific target beneath it (all states)
- **Someone Joined**: Another UAV has joined this UAV in monitoring its current target (Track, Assess Damage, Attack)
- **Attack Begun**: Another UAV has just begun an attack on this UAV’s current target (Assess Damage, Attack)
- **Attack Succeeded**: An attack on the UAV’s current target has successfully destroyed it (Assess Damage)
- **Attack Failed**: An attack on the current target failed to destroy it (Assess Damage)
- **Alone**: The UAV is in the Attack state but there are no other UAVs performing damage assessment (Attack)

The behavior of each UAV depends on its state-transition rules, which conceptually represent a state-transition diagram. The default state-transition diagram is represented in Figure 6. Each node in the diagram represents a state, while the directed edges represent probabilistic transitions between states caused by the reception of a given signal.

![State transition diagram for switching between behaviors.](image)

**Figure 6**: State transition diagram for switching between behaviors.
Experimental setup and performance metrics

We have run thousands of simulations under various combinations of the following parameters:

- Number of UAVs: 4, 8, 16 or 32
- Number of targets: 4, 8 or 16
- World size: 1224x1224, 1500x1500, 1732x1732 (note that these dimensions are such that the areas are in a ratio of 1.0-1.5-2.0)
- Duration of the mission: 1800 or 900 seconds
- Drop point: all UAVs are dropped from one of eight points evenly spaced around the edge of the world (0=SW, 0.125=S, 0.25=SE, 0.375=E, and so on)

In most cases reported below, every data point is the average of twenty runs, each run with a different random seed. In a few cases, we only averaged over ten runs, though we found that this was generally a sufficient number of runs to obtain a good estimate.

We established several ways of measuring performance under various simulation conditions:

- Overall kill probability (PK): what percentage of targets are killed during a mission?
- Number of UAVs destroyed (DU): UAVs can only be destroyed as a result of an attack.
- Probability of killing target i (PK_i): over a set of simulations with identical conditions but varying random seeds, how often (as a percentage) was target i successfully destroyed?
- Time to kill target i (T_i): how much time elapsed before target i was destroyed.

Note that for both T_i and PK_i, the index of the target refers to the order in which the target was destroyed. In other words, each number is not associated with a particular target: whichever target happened to be destroyed first, it is target 1, and so on. Hence PK_i refers to the probability that i targets have been destroyed, and T_i refers to the time that it took to destroy i targets.

Two points should be noted for all results shown. (1) the simulation is capped at 1800 seconds. Any target not killed by the end of the simulation has its T_i set to 1800. Hence all time plots are capped at 1800. (2) Because of the recruitment strategy, the last surviving UAV will never attack (we realize this should eventually be changed). Hence a simulation with N UAVs will never result in more than N-1 targets being destroyed.

Suppression Mission Results: PK

We first look at cumulative PK, that is, the percentage of targets killed during the course of a mission. Figure 7 shows how the PK varies as a function of UAVs for various numbers of targets, and as a function of the number of targets for various numbers of UAVs.

![Cumulative PK](image)

Figure 7: Cumulative PK as a function of the number of UAVs and targets. Each data point is averaged over 20 runs each at all three different world sizes.

The results in Figure 7 are averaged over three world sizes. Starting from the upper figure, we see that increasing the number of UAVs with a fixed number of targets increases the overall PK. While this is not surprising, what is interesting is the shape of each curve, and how this shape changes as a function of the number of targets. For instance, the fact that the curve is more S-shaped for a larger number of targets suggests that the effectiveness of increasing the size of the swarm depends on how large the swarm already is. This type of plot can be used to answer questions such as “how many UAVs do I need to send to an area in order to guarantee that 80% of targets are destroyed during a half-hour mission?”
Figure 8: Probability of killing 1-4 targets as a function of swarm size.

The lower graph in Figure 7 shows essentially the same data, but plotted as a function of the number of targets. This type of plot answers questions such as: “given that I have a swarm of N UAVs, how many targets will they be able to destroy with an 80% probability within a half hour mission?”

An interesting point about these results is that we have found that the world size is largely irrelevant, especially with the largest number of targets. What this suggests is that the effectiveness of a swarm is related to the target density, and that there exists a critical target density, beyond which the swarm is likely to perform equally well. In other words, all three world sizes are such that putting in 16 targets “saturates” the world, even at the largest size.

**Suppression Mission Results: PK**

Let’s now look at how the PK evolves as a function of the number of targets. Figure 8 shows the average PK for \( i \) targets, where \( i = 1, 2, 3, 4 \). In other words, we lump in all cases with 4, 8, or 16 targets, and all three world sizes, and we ask the questions: “what is the probability that within 1800 seconds I will kill one target? two targets? three targets? four targets? And how does this probability vary with swarm size?”

Some points are worth discussing here. First, as mentioned before, there is no way that 4 UAVs could kill 4 targets, so \( PK_4 \) should be zero for 4 UAVs, as is in fact observed. Second, it is interesting to see that the lines seem to be similar but shifted to the right. This suggests that for any swarm size there tends to be fairly flat performance near 100% until a certain target number is reached, at which point performance declines rapidly. This again suggests that it may be possible to calculate explicitly the optimal swarm size to use for a given problem. For instance, from Figure 8 we would suggest that for all world sizes considered, if the goal were to destroy two targets, there would be no advantage in sending 32 UAVs instead of 16.

**Suppression Mission Results: TK**

We now shift our attention to a consideration of the time required to achieve a certain goal. In particular, rather than wanting to know how likely it is that we will kill \( i \) targets in a fixed amount of time, we may want to study how long it takes to kill \( i \) targets, on average. Figure 9 shows results along these lines.

These results make it possible to answer questions such as: “If I send in 8 UAVs in a 10-squared-mile area, how long will it take me to clear out four targets?”

One important conclusion we can draw from the results shown in Figure 9, which supports the conclusions drawn from earlier results, is that even when time is the key constraint, it is possible to extract useful information about an ideal swarm size to be using. In the case of Figure 9, which focuses on simulations with a world measuring 1224x1224 units, it is clear that there is a huge benefit in going from 4 to 8 UAVs. The benefit is not as pronounced in going to 16 UAVs, and is almost completely gone in changing from 16 to 32 UAVs.
**Figure 9:** Average time required to kill \( i \) targets as a function of swarm size.

**DISCUSSION**

We have presented some results with an agent-based model of decentralized control strategies for swarms of UAVs. This work lays the foundation for research that will be an essential ingredient for the successful deployment of UAV swarms. We believe that most research on UAV swarms has been qualitative in nature. A systematic approach such as the one outlined here will be critical in order for swarm control strategies to be adopted widely.

We have been able to show that even some fairly simple control strategies based on local communication can yield satisfactory results on search or suppression missions. More importantly, we have shown how one might begin to answer quantitative questions about the functionality, scalability, and robustness of UAV swarms.

The idea of using decentralized control strategies for UAV swarms has been described in only a few other publications. We summarize here two examples that are somewhat closely related to our own work.

Parunak and colleagues$^{4,5}$ have proposed to use digital pheromones to control UAV swarms. Specifically, they proposed to cover a terrain with a grid of “place agents,” which could be physically implemented as ground sensors. These sensors distribute information among themselves about threats and targets, and also interact with UAVs, which are represented as “walker agents.”

The digital pheromones consist of signals representing threats, targets, and other characteristics. These signals are stored by individual place agents, they can diffuse to neighboring place agents, they can evaporate, and they are used by walker agents as a basis on which to decide where to go at each time step. Parunak $et al.$ demonstrate that, under certain assumptions, the diffusion and evaporation of pheromone results in a representation akin to potential fields, a well-known method for autonomous navigation in the presence of attractors and repulsors.

Using their digital pheromone approach, these authors test performance on a variety of missions, with UAV swarms of up to 100 units, and with a variety of sensory and/or weapon configurations. The results we have seen are not sufficiently detailed to allow for a careful analysis, but at least superficially they seem promising. One surprising omission, at least in the articles we were able to access on-line, is that the performance comparison is not normalized by the size of the swarm. In other words, Parunak $et al.$ compare directly the results (be they in terms of target identification or destruction) with 10, 50 and 100 UAVs, and with various configurations. Not surprisingly, larger swarms perform better than smaller ones.

On the positive side of the equation, Parunak $et al.$ include details about missions that we completely
overlook in our own simulations (such as specific target, UAV, and threat types) and they have even integrated their approach with existing platforms, such as EADTB. Other strong points of their work are: the ability to adapt dynamically through the use of “ghost” agents; and the ability to create paths to targets that are partially surrounded by threats—a classical problem that gradient-based navigation schemes are unable to solve.

Nygard and colleagues\(^6\) have used agent-based modeling principles to control intelligent flying munitions. Their goal is to coordinate multiple autonomous munitions with an approach that allows for adaptation to unexpected changes during a mission, such as the appearance of threats.

Altenburg et al. design a system that includes agents, the environment, and communications mechanisms. Agents are endowed with the ability to perform several behaviors: avoidance, attraction, following, dispersion, aggregation, homing and flocking. These behaviors are triggered and modulated through internal and external signals.

The authors describe a Java simulation tool they developed, and some preliminary experiments in which desirable behaviors emerge from local interactions and control rules. The cited article shows an example in which multiple UAVs try to coordinate a strike from multiple directions. A more recent example, presented at an ONR meeting in July of 2002, showed a more complex missions similar to the search-and-suppress mission we are studying. In that presentation, Dr. Nygard showed a team of UAVs flying over a search area using a series of waypoints just outside the edge of the search area itself. By defining local rules for the UAVs, the swarm as a whole could carry out the mission under a variety of configurations. For instance, if one of the UAVs detected a high-priority target and immediately destroyed it, the other UAVs would automatically reconfigure their flight pattern at the next waypoints so as to ensure uniform coverage during the rest of the mission.

While the results are interesting, we feel that this research is limited in that it does not easily generalize to other configurations: the particular decentralized control strategy that yields the desired swarm-level behavior was handcrafted; changing the design to accommodate different mission parameters or constraints would require a manual modification of the decentralized rules. This stands in contrast to our approach, in which we leverage evolutionary design and other aspects of swarm intelligence to design local, decentralized control strategies that yield a desirable global behavior.

**BIBLIOGRAPHY**


