Swarm Intelligence: a New C2 Paradigm with an Application to Control of Swarms of UAVs

Paolo Gaudiano, Benjamin Shargel, Eric Bonabeau
Icosystem Corporation
10 Fawcett St.
Cambridge, MA 02138

Bruce T. Clough
Air Force Research Laboratory
Control Sciences Division
Wright-Patterson AFB, OH 45433

Swarm Intelligence: a New C2 Paradigm with an Application to Control of Swarms of UAVs

Paolo Gaudiano, Benjamin Shargel, Eric Bonabeau
Icosystem Corporation
10 Fawcett St.
Cambridge, MA 02138

Bruce T. Clough
Air Force Research Laboratory
Control Sciences Division
Wright-Patterson AFB, OH 45433

Abstract

We have developed an agent-based model to simulate command and control of a swarm of Unmanned Air Vehicles (UAVs). Our approach makes use of decentralized strategies to control a UAV swarm carrying out a search mission. In this paper we introduce our approach, we present the details of the proposed model, and we provide results of simulations testing our control strategies under a variety of configurations. We also outline some initial results obtained by extending our simulation tool to include the ability to carry out missions in which UAVs can track moving targets, strike targets, and perform Battle Damage Assessment.

Introduction

The DoD is investigating the development of swarms 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 controlling UAVs, which rely on centralized control and on the availability of global information, are not suited to the control of UAV swarms, 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 (Bonabeau, Dorigo and Theraulaz, 1999). 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. Social insects in particular provide us with a powerful metaphor for designing collectively intelligent systems comprised of a number of agents. These agents not only process information but also perform actions that change their internal state, their environment, and the environment of the other agents. Despite noise in the environment, errors in processing information and performing tasks, and a lack of a global communication system, social insects are very efficient at performing group-level tasks.

This paper presents initial findings of novel UAV modeling in swarm conditions and discusses the extensibility to other sensor and C2 problems that may benefit from decentralized approaches to military command and control in network-centric environments. In this research, UAVs are controlled through local rules, but attempt to achieve a common goal as a swarm. We devised some control strategies based on strictly local information, and other strategies that involve varying degrees of global coordination. Performance was tested systematically under a variety of control strategies and configurations. The simulator was then extended to allow UAVs to track moving targets, strike targets, and perform Battle Damage Assessment (BDA).

Simulation of a UAV search mission

We have developed a simulation tool to study the behavior of a UAV swarm carrying out an area search mission. A cluster of UAVs is dropped somewhere along the perimeter of a square search area. Several static targets are located within the search area. The UAVs disperse through the area and try to sweep over as much of the area as possible to locate the targets. As long as the targets are stationary and positioned randomly within the search area, the mission could be defined equally as a target search mission or as an area coverage mission. In a later section we describe preliminary results with an extension of this model that includes moving targets, and in which UAVs can strike the targets and perform BDA.

Our 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. The sensor is stochastic, with the probability of detecting a target dependent on distance, elevation, and the amount of time spent flying over a given terrain cell.
  - 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 in the area-coverage version are static, randomly distributed throughout the region (see later sections for an extended version with moving target and additional UAV functionality and communications).

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 is a screenshot of the initial version of the UAV simulator. 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 (blue 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."

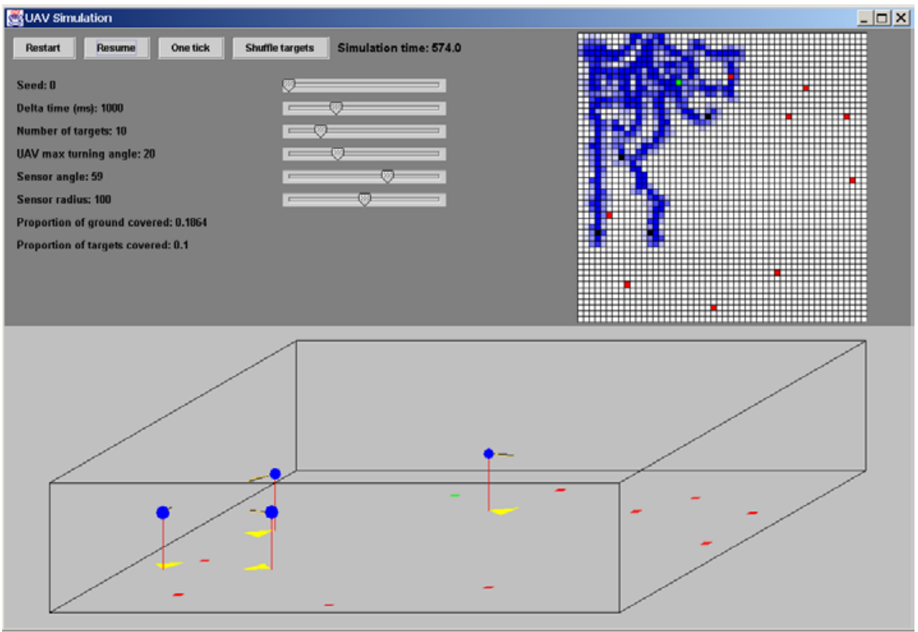


Figure 1: Screenshot of the UAV simulator.

Finally, the top-right area 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) 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.

Quantification of resultss

In order to compare different control strategies and different parameter settings it is crucial to develop a systematic, quantitative approach. For the results that follow we ran simulations varying key parameters over useful ranges. At each particular condition, we ran at least five simulations with different random seeds. Each data point in the results reported below represent the average over the five runs.

The results reported below focused on variations in two primary factors:

- number of UAVs (1-10, with an additional series of runs with up to 110 UAVs);
- UAV control strategy (we tested five strategies individually and in combinations); some strategies were tested at different parameter levels when appropriate.

Aside from these factors, we tested variations of many other parameters, such as UAV maximum speed, size of the terrain, duration of the simulation, UAV dynamic control parameters (steering rate, wall avoidance behavior, ...). The results presented in later sections are based on what seem to be fairly “stable” parameters that gave meaningful results.

UAV 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.
- 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 regions, and that a central controller monitors the level of coverage within each region, as well as the number of UAVs currently in that region. As we will discuss later, we tried several variants of the global strategy.

The baseline “strategy” is used as a comparison for the other strategies.

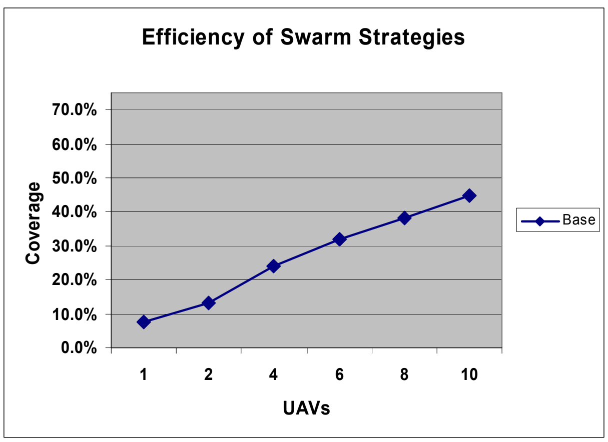
Summary of area-coverage simulation results

For all the 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. More detailed results can be found in our progress reports.

<u>Baseline strategy</u>

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.



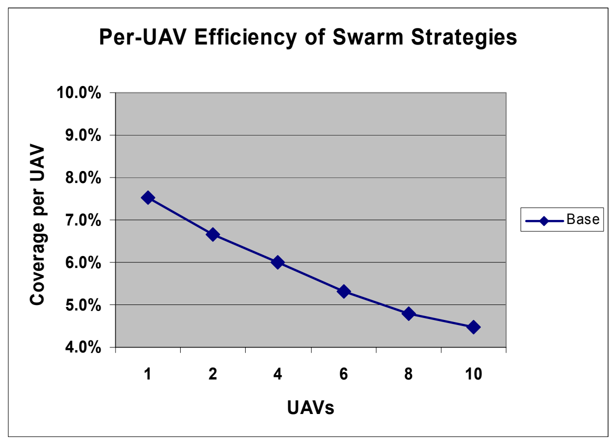


Figure 2: Coverage efficiency in the baseline case.

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.

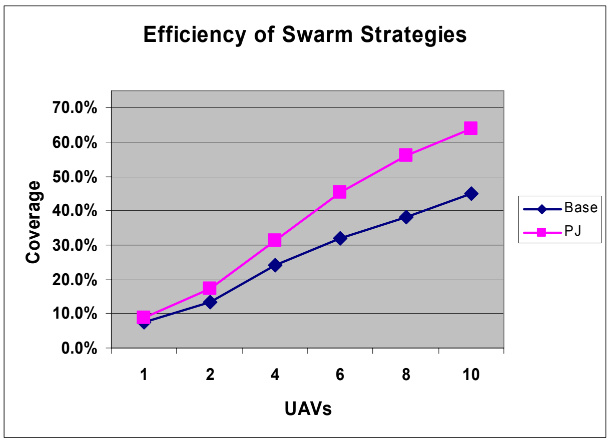
<u>Testing additional strategies to improve coverage</u>

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.

When we combine jitter and repulsion with the pheromone strategy, the performance of the swarm improves significantly. Figure 3 shows the quantitative results obtained with this combination of strategies.



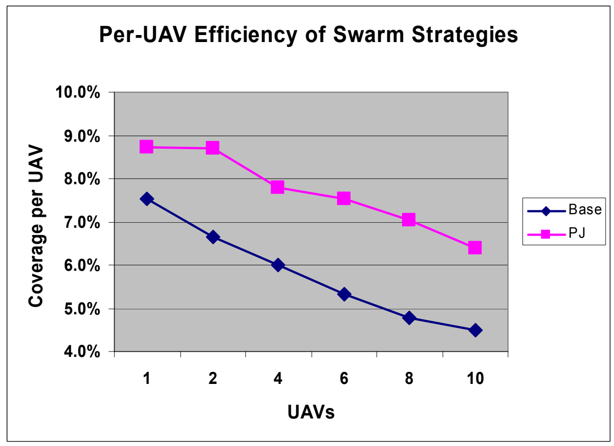


Figure 3: the use of a local "pheromone" strategy yields superior results.

We have found that the pheromone strategy is the crucial element to the superior performance shown here. Figure 4 reinforces this point by showing the behavior of UAVs during a single run. The matrix on the left is the coverage trace of a run with 10 UAVs and a 60-unit repulsion radius. The matrix on the right 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.

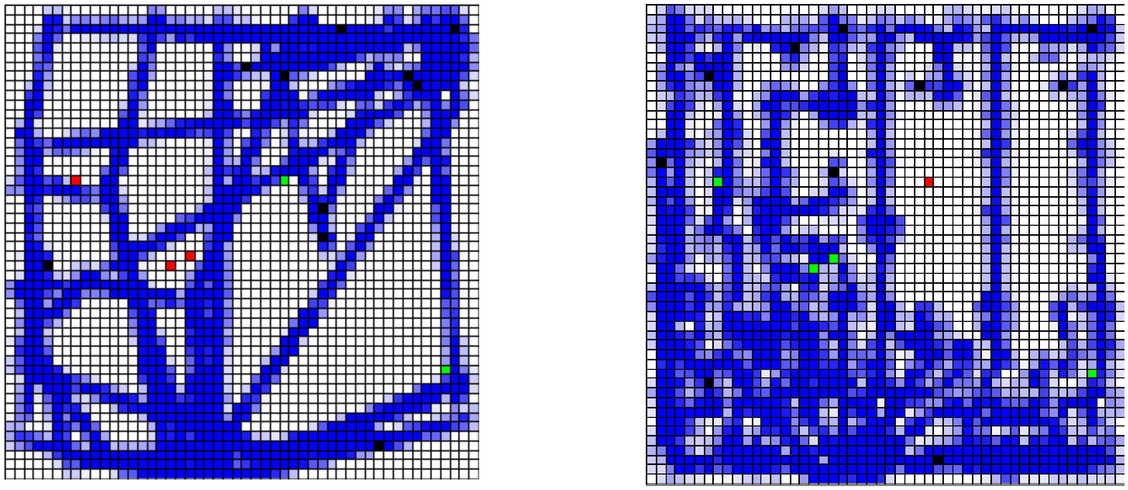
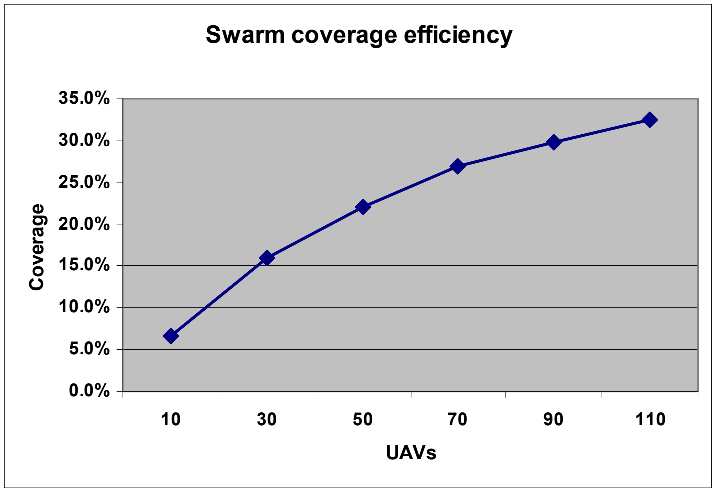


Figure 4: Coverage matrices resulting from repulsion (left) and pheromone (right) strategies.

It is 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.



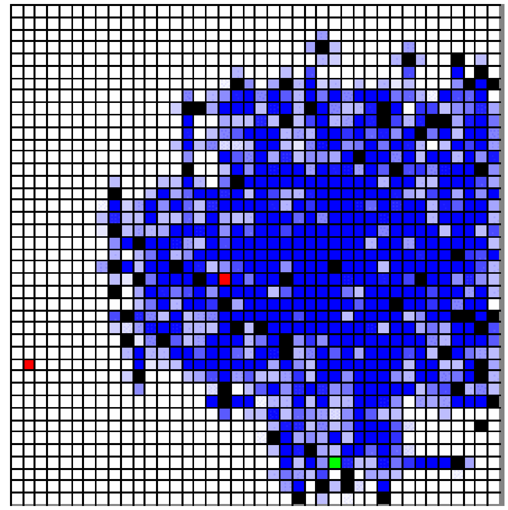


Figure 5: Coverage as a function of swarm size for larger swarms.

The left side of 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). The right side of the figure shows one sample run with 70 UAVs. Note that because of the size of the area, each cell in the matrix covers several hundred units of displacement, so that a single black square may be marking the location of multiple UAVs.

Extending the simulator to model suppression missions

In this section we summarize some preliminary results obtained with an extended simulator in which targets move randomly over the search area, and UAVs are able to track and attack the targets. Full details will be published in a separate manuscript.

In order to progress towards more realistic mission scenarios, the goal of the UAVs is now not to maximize coverage of the target area but strike 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 changes we made to the model in order to carry out the new type of mission:

- UAVs are able to strike targets by diving into them. The UAV is destroyed in an attack regardless of whether or not the strike was successful.
- UAVs carry sensory payload that enables them to perform BDA after another UAV has attacked a target.
- 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

- Four new UAV behavioral states have been added to the original Search state: Join, Track, Assess Damage and Attack. The current state of a UAV determines how it updates its velocity vector each time step, in conjunction with "always on" wall-avoidance and height-maintenance behaviors.
    - The **Search** state is the equivalent to what was used for the area-coverage mission described earlier: the UAV aims for unexplored regions of the world, according to local and/or global exploration gradients.
    - In the **Join** state, a UAV aims for the coordinates of another UAV which has recruited it.
    - In the **Track** state, a UAV tries to remain above its current target at a specified tracking height.
    - The **Assess Damage** state is similar to the Track state, but the UAV is also focusing on the condition of the target to determine whether or not another UAV's attack has disabled it.
    - In the **Attack** state, a UAV plummets to the ground to destroy its current target. The success of an attack is stochastic, and could be modified in the future to take any relevant information (such as target type and environmental conditions) into account.

Summary of results

In this section we report our results measured in terms of cumulative *Probability of Kill* (PK), that is, the percentage of targets killed during the course of a mission. Figure 6 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.

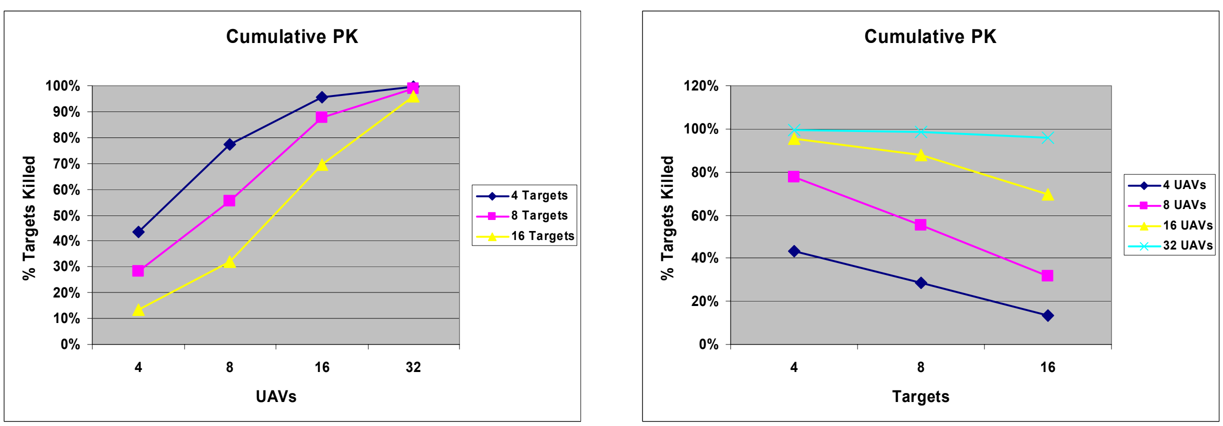


Figure 6: Cumulative PK as a function of the number of UAVs and targets. Each data point is averaged over 20 runs each at all three different world sizes.

These results in Figure 6 are averaged over 20 runs and three different world sizes: 1224x1224, 1500x1500 and 1732x1732 (note that these dimensions are such that the areas are in a ratio of 1.0-1.5-2.0). Starting from the left 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?" The graph on the right of Figure 6 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 observation is that the shape of the curves above does not seem to depend heavily on the size of the world. This is clear in Figure 7, which plots the same data, but broken out by world size.

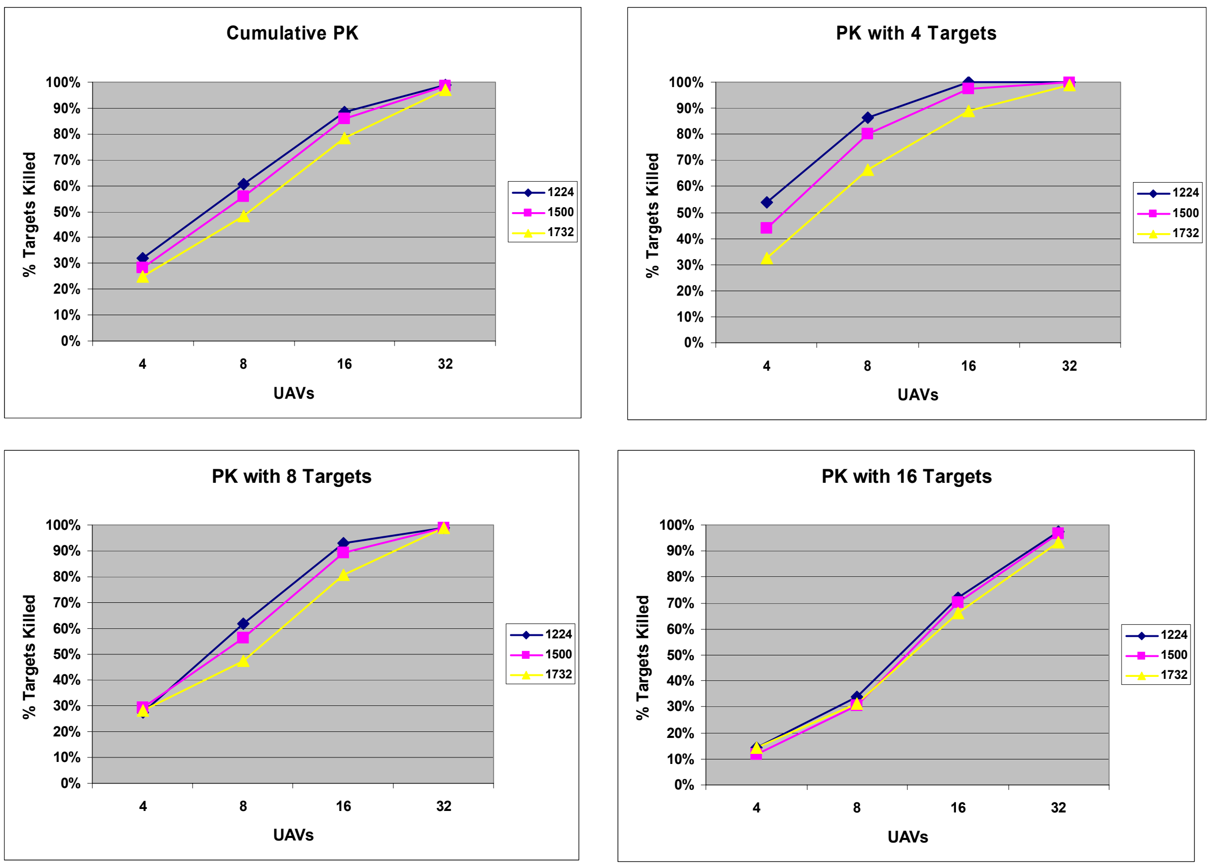


Figure 7: Cumulative PK as a function of swarm size, broken out by world size. **(a)** Results accumulated over all target numbers (4, 8, 16); **(b-d)** results obtained with 4, 8 16 targets only.

What appears from the results of Figure 7 is that the world size seems 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.

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 (Parunak, Purcell and O'Connell, 2002; Sauter, Matthews, Parunak and Brueckner, 2002) 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 decided 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 have used agent-based modeling principles to control intelligent flying munitions (Altenburg, Schlecht and Nygard, 2002). 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

- Altenburg, K., Schlecht, J. and Nygard, K.E. (2002). An agent-based simulation for modeling intelligent munitions. In *Proceedings of the Second WSEAS Int. Conf. on Simulation, Modeling and Optimization*. Available at www.cs.ndsu.nodak.edu/~nygard.
- Bonabeau, E., Dorigo, M. and Theraulaz, G. (1999). Swarm Intelligence: From Natural to Artificial Systems. New York, NY: Oxford University Press.
- Parunak, H.V.D., Purcell, M. and O'Connell, R. (2002). Digital pheromones for autonomous coordination of swarming UAVs. *Proceedings of the AIAA's First Technical Conference and Workshop on Unmanned Aerospace Vehicles, Systems, and Operations*. **AIAA 2002-3446**.
- Sauter, J. A., Matthews, R., Parunak, H.V.D. and Brueckner, S. (2002). Synthetic Pheromone Mechanisms for Coordination of Unmanned Vehicles. *To appear in Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*.