

INVESTIGATIONS OF DDDAS FOR COMMAND AND CONTROL OF UAV SWARMS WITH AGENT-BASED MODELING

Ryan McCune
Rachael Purta
Mikolaj Dobski
Artur Jaworski
Greg Madey

Yi Wei
Microsoft Corporation
1 Microsoft Way
Redmond, WA 98052, USA

Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46656, USA

Alexander Madey

Trinity School at Greenlawn
107 South Greenlawn Avenue
South Bend, IN 46617, USA

M. Brian Blake

Department of Computer Science
University of Miami
Miami, FL 33124, USA

ABSTRACT

The application of Dynamic Data Driven Application Systems (DDDAS) to the command and control of swarms of Unmanned Aerial Vehicles (UAVs) is being investigated. Swarm intelligent systems are not only efficient at solving group-level problems, but also decentralized, controllable by few simple parameters, making possible the command and control of UAV swarms by a single operator. Four separate but related projects are surveyed that explore the command and control of UAV swarms. Each project employs the DDDAS paradigm, entailing the ability of an executing application to incorporate dynamic data into the decision process, and conversely, to steer the measurement process via a central application system. By providing an overview of DDDAS approaches to UAV swarm mission scheduling, UAV swarm communication, UAV swarm formation planning, and flocking applications, general principles of UAV swarms and DDDAS architecture may be observed.

1 INTRODUCTION

The applications of Unmanned Aerial Vehicles (UAVs) have proliferated with recent technological advancement. Since UAVs do not require an on-board pilot, vehicles are less expensive than piloted counterparts and offer greater design flexibility, such as smaller size and increased mobility. Control is administered either autonomously by an on-board computer, by one or more remote ground operators, or by a combination of both. Driven by increased accessibility, UAVs are now utilized across a variety of domains, including real estate, agriculture, police surveillance, and parcel delivery (Hodson 2013). Military applications include reconnaissance of unknown territories, monitoring hazardous environments, and opportunities for minimizing risks to safety of personnel.

As costs reduce and capabilities improve, coordinated teams of UAVs will likely supplant single UAVs for more complex applications (Cambone 2005). However, current control methods limit the effectiveness

of such teams. Each UAV requiring one or more highly-trained ground pilots does not scale well as the number of airborne UAVs increases. Centralized control approaches frequently lead to exponential increases in communication bandwidth requirements and software complexity (Gaudio et al. 2003). To solve problems cooperatively while maintaining scalability, application designers are investigating swarm intelligent methodologies for teams of UAVs.

Swarm intelligent systems are characterized by emergent problem solving capability, where simple behaviors of individual agents give rise to complex phenomena, rendering the whole greater than the sum of parts. Examples are commonly cited in nature, such as the flocking of birds, or ant colonies foraging the shortest path to food (Bonabeau, Dorigo, and Theraulaz 1999). Swarm control is decentralized, with communication performed locally, so the swarm is both scalable and robust. Agents can be added to the swarm with minimal overhead, and the failure of any single agent is of minimal detriment to global performance.

Swarm intelligent behaviors overcome current challenges with UAV teams by offering efficient, cooperative problem solving facilitated by decentralized control. Previously, our research group identified three challenges to the implementation of swarm command and control for UAVs (Madey et al. 2012). These challenges are (1) the need for near real-time dynamic command and control of the swarm, (2) efficient mission planning and dynamic real-time retasking of the swarm, and (3) the need for improved automation of swarm mission planning and command and control. This paper summarizes our previous work addressing these challenges from different focuses (Wei, Madey, and Blake 2013; Purta, Nagrecha, and Madey 2013; Madey and Madey 2013; McCune and Madey 2013).

All projects are part of an on-going investigative effort, sponsored by the Air Force Office of Research, to explore effective modeling and control methods for operating future UAV swarms within a Dynamic Data-Driven Application System (DDDAS) (Darema 2004). DDDAS entails the ability of an executing application to incorporate dynamic data into the decision process, and conversely, for the application to dynamically steer the measurement process. A dynamically controlled measurement process means DDDAS supports continual simulation calibration. Since swarm models exhibit non-linear dynamics stemming from unpredictable agent-agent interaction, incorporating swarm simulations into a framework allowing repeated simulation calibration will improve simulation accuracy, making swarm modeling within DDDAS a compelling topic of investigation.

The rest of this paper is organized as follows. In Section 2, a brief introduction to the background of DDDAS and UAV swarm control and operation is given. Section 3 starts with an overview of the swarm project, followed by detail descriptions of each individual sub-projects. In Section 4, we summarize our research and give a brief preview of future works.

2 BACKGROUND AND RELATED WORK

Projects surveyed in this paper apply principles of DDDAS to the command and control of UAV swarms. Each project is designed to incorporate an agent-based simulation into a UAV swarm scenario. Agent-based modeling and simulation is an intuitive paradigm for modeling swarm intelligent systems. Background is provided on swarm intelligence, agent-based modeling and simulation, and DDDAS, then work related to the surveyed projects is presented.

2.1 Background

2.1.1 Swarms

Though the term "swarm" may often refer to a specific biological instance of swarm intelligence, such as a swarm of bees, a swarm may also more generally refer to any system exhibiting swarm intelligence. Swarm intelligence characterizes a system of simple, interacting agents that collectively exhibit complex phenomena, where the whole is greater than the sum of the parts. Two common examples are observed in nature. The flocking of birds can be modeled as a swarm, where the collective flock formation emerges from

each bird adjusting direction and velocity based solely on neighboring birds (Reynolds 1999). Similarly, ants in ant colonies communicate indirectly through pheromones, a mechanism through which the shortest path to food is uncovered while foraging (Bonabeau, Dorigo, and Theraulaz 1999). Both examples illustrate decentralized control, where agents communicate only locally while collectively accomplishing a global task. Decentralization supports system scalability, since agents can be added without creating communicative bottlenecks. Moreover, decentralized systems are robust, since the failure of a single agent will not cascade throughout the system. The emergent problem solving ability of swarms contrasts with traditional, centralized approaches to problem solving.

2.1.2 Agent-Based Modeling

Agent-based modeling and simulation is an approach to representing a system as autonomous agents that interact amongst one another as well as with the environment. Agent behavior is designed to capture local interactions, through which, after repeated interaction, the collective characteristics of a system emerge over time. Agent-based modeling is an intuitive paradigm for representing swarms. Modeling a system at the agent-level is considered a bottom-up approach, in contrast to conventional top-down models, like equations. Notably, the bottom-up approach can capture the generative nature of system properties, leading researchers to consider agent-based simulation as "a third way of doing science," (Axelrod 2006). While agent-based models offer tremendous explanatory capability, the rapid interaction between agents can quickly lead to unpredictable, non-linear results, requiring careful model calibration. Model verification and validation also challenge the paradigm.

2.1.3 DDDAS

DDDAS entails the ability of an executing application to incorporate simulated data into the decision process, while conversely being able to dynamically manage sensors to refine measurements (Darema 2004). New sensor data is assimilated into the running simulation, enabling a synergistic feed-back and control-loop between the real-world application and the simulation model. Synergy is realized through a simulation capable of modeling complex, non-linear dynamics in faster than real-time. Sensor controls continually drive the measurement process in order to recalibrate the simulation and provide precise, meaningful results. Since agent-based simulations often require detailed calibration to avoid unpredictable dynamics, incorporating an agent-based model within a DDDAS framework minimizes model inaccuracy by repeatedly recalibrating with new data. An accurate model can support application optimization via simulation.

2.2 Related Work

Challenges imposed by UAV swarms have attracted much research attention, including new simulation tools, simulation models, and command and control mechanisms to address all aspects of swarming. Previous simulation work has utilized MASON (Luke et al. 2005), a general purpose multi-agent simulation library for Java, as well as the Matlab-based UAV simulator MultiUAV2 (Rasmussen et al. 2005). In (Garcia and Barnes 2009), a multi-UAV simulator is implemented on top of a commercial flight simulator called X-Plane, and another tool is presented in (Russell, Lamont, and Melendez 2005) that utilizes an existing parallel emulation and simulation tool called SPEEDS. Agent-based models of search and destroy strategies are presented in (Gaudio, Bonabeau, and Shargel 2005). Similarly, cooperative search is discussed in (Varela et al. 2011). Path planning and vehicle routing problems are explored using multi-objective evolutionary algorithms in (Lamont, Slear, and Melendez 2007). Another decentralized strategy for command and control of UAV swarms is modeled in (Gaudio et al. 2003). DDDAS is applied to multi-UAV scenarios in (Khaleghi et al. 2013).

3 TECHNICAL APPROACHES

3.1 Overview

We discuss four projects that investigate swarm control methods with agent-based models. While synthesized swarms aim for fully distributed and decentralized systems, UAV swarms require an element of control from a central operator, no matter how minimal. Each project adapts a swarm behavior to a UAV mission and explores the role of a central controller. The application of the agent-based model to the DDDAS framework is also discussed.

The first project explores mission scheduling within the swarm. The second project augments two swarm algorithms with an additional behavior to optimize the search for mobile targets. The third project compares swarm performance based on communication. The fourth implements two swarm scenarios directly applicable to the DDDAS framework.

3.2 Swarm Mission Scheduling

One of the interesting and challenging problems in UAV swarm control and navigation is mission scheduling. In our study, a long endurance swarm is considered. The swarm is capable of carrying out multiple missions, as well as accepting new missions on the fly. The goal then, is that when given a swarm S , either with or without existing missions, and a new mission MS , how to schedule the mission onto the swarm so that the total mission cost is minimized.

It is assumed in our study that a mission consists of multiple tasks. And a task can be finished by any UAV. The scheduling process produces a mapping, or an assignment, M , from the set of tasks T to the set of UAVs, S . The mapping specifies which task is assigned to which UAV. For example, $\{t_1, v_2\}$ means UAV 2 gets task 1.

The mission cost is the sum of all its tasks' costs. The cost of a task is different for different types of tasks. Generally, it consists of the traveling cost and the task completion cost. The former is the cost for a UAV to travel to the task location from its current location, while the later is the cost for the UAV to carry out the task.

3.2.1 Global-Local Hybrid Planning and Scheduling

To solve this problem, a hybrid approach is employed which combines global task assignment and local task scheduling. Two types of agents are used. One is the *Swarm Control Agent* (SCA), and the other is the *UAV agent* (UA). A swarm operation environment includes a single SCA and multiple UAs. The SCA is the interface between the human operator and the swarm, it usually runs in the ground station and its design objective is to decompose a newly arrived mission into multiple tasks, assign those tasks to UAVs, and monitor the status of the swarm and missions. On the other hand, a UA represents an operational UAV in the swarm which is capable of completing different tasks. Agents communicate with each other through messages. These messages either require the recipient to take an action, or contain the latest information about the sender. Six types of messages are used by the agents. For a complete description of these messages and their purposes, refer to Wei et al. (Wei, Blake, and Madey 2013).

The SCA is also delegated with the tasks to monitor task dependencies as well as mission and swarm status. It needs to decide when a task is ready to be assigned. A ready task is the one that doesn't have any predecessor tasks or all of its predecessors have been finished. As a task is completed, other tasks that are dependent on it may become ready. For each ready task, the SCA selects the assignment target UAV by the following steps:

1. Calculate, for each UAV, the cost of finishing the task. The SCA will use its latest knowledge about the swarm to estimate the cost of finishing it. This cost is different for different UAVs, because UAVs are on different locations and may or may not have existing tasks. There are two situations:

- (a) The UAV doesn't have scheduled tasks. In this situation, the cost is the traveling cost from UAV's current location to task location plus the task's completion cost. The capacity used for comparison is the UAV's current residual capacity;
 - (b) There are scheduled tasks on the UAV already. In this case, the estimated final location and estimated final residual capacity values are used to calculate the task cost.
2. Sort the calculated costs and choose the UAV with the smallest cost value as the candidate for the task.

The task assignment gives a task to a UAV, but it does not specify how this task should be scheduled on the UAV. To be more specific, the SCA does not specify whether the new task should be executed immediately or after all existing tasks are completed. The UAV controls this decision. Several policies are implemented to carry out this local task scheduling process. A sorted task list is produced on each UAV after the scheduling process, which represents the completion order of the tasks. The UAV then follows the task list to complete all tasks.

The proposed swarm mission scheduling framework is implemented as a Java multi-threaded simulation program. Both the SCA and all UAs are represented by threads. Figure 1 shows the GUI of the simulation. The left part of the GUI has the mission output area on top and the control buttons at the bottom. The testbed visualizes missions by displaying their tasks and task dependencies as DAGs. Each task node also has a color indicating the task status.

At the bottom left of the GUI are the control buttons. The Start button starts a new simulation. The Stop button terminates a running simulation. The Pause button is used to pause/resume the simulation.

The top area of the GUI's right part is the swarm display area. The swarm is visualized in this area, including all its UAVs, their default waypoints and all tasks scheduled on these UAVs. A UAV is represented by a small solid triangle on the GUI. The capacity of the UAV is shown by a small solid rectangle near the UAV. The area below the swarm display area is used for textual output of simulation information, such as task assignment results, task and mission completion results, and simulation termination message. For more detailed descriptions of the simulation and all the experiments, please refer to Wei et al. (Wei, Blake, and Madey 2013).

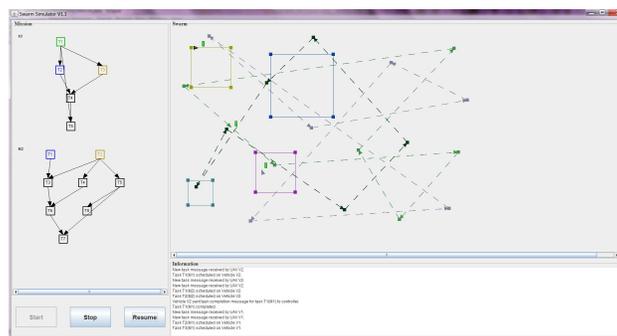


Figure 1: The GUI of the simulation application, displaying a swarm of 3 UAVs executing 2 missions (Wei, Madey, and Blake 2013).

3.3 Formation Planning

Swarm intelligent systems utilize simple local behaviors to collectively solve complex problems, where the capability of the group is greater than the sum of its parts. Emergent phenomena in swarms, the mechanism to collectively solve problems, often occur non-linearly and may be difficult to predict. For real-time systems utilizing swarm intelligence, modeling possible control decisions using agent-based simulation can

improve performance. We developed a simulation that can be implemented within a DDDAS framework to support search formation assignment (McCune and Madey 2013).

Adapting swarm methods to relevant problems in the UAV domain is challenging. One recently problem proposed is Cooperative Hunting (Altshuler, Bruckstein, and Wagner 2005), where agents work together and search an environment in such a manner as to guarantee all intelligently evasive targets are found in a finite amount of time. Their work introduced two swarm search algorithms, the Parallel Path algorithm and the SWEEP Protocol, depicted in Figure 2 and Figure 3, respectively. Both algorithms implement agent formations in such a way that a mobile target, with a predetermined speed, cannot avoid detection.

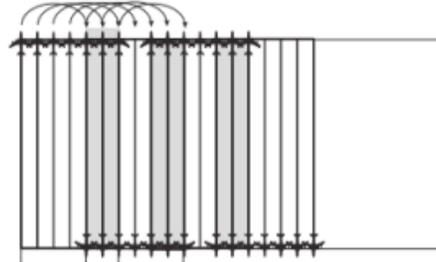


Figure 2: The Parallel Path algorithm, where agents sweep back and forth in parallel to ensure all mobile targets are found. The shaded area represents overlapping paths (Altshuler et al. 2005).



Figure 3: Execution of the SWEEP Protocol over time. Agents are placed along the perimeter of a search space, sweeping the outside edges moving inward (McCune and Madey 2013).

The two algorithms come with trade-offs. Parallel Paths works as a simple sweep back and forth. The formation is easy to implement and requires little communication between agents other than coordinated movement. However, in current form, the formation is only able to effectively search rectangular environments. Conversely, the SWEEP Protocol is capable of searching any connected shape when provided enough agents, but at the expense of significant increases in complexity. The SWEEP Protocol contains intricate rules, can be difficult to implement, and requires considerably more communication and computational power, since an agent must determine whether or not its location keeps the search space connected (Altshuler and Yanovski 2005).

With these trade-offs in mind, our project implements an additional behavior, the Sentry agent, that allows for the search space to be partitioned. The Sentry is a stationary agent that patrols a single location, equipped with a stronger sensor capable of searching with a larger radius. A central controller can assign Sentries to locations that partition the search space into sub-regions, depicted in Figure 4. Considering the trade-offs between each formation, the number of agents and the agents formation can be selected to optimize particular parameters, such as time, communication, or robustness. Furthermore, UAVs may fail over the course of a mission, necessitating formation reassignment. The simulation gives the operator a tool to test if certain formation assignments will successfully complete the mission.

The addition of the Sentry adds flexibility to formation assignment in the Cooperative Hunters problem. While the search swarms act deterministically and autonomously, assignment comes from a central controller. Agents in a UAV swarm may fail, so a simulation tool is helpful for dynamic reassignment during a mission.

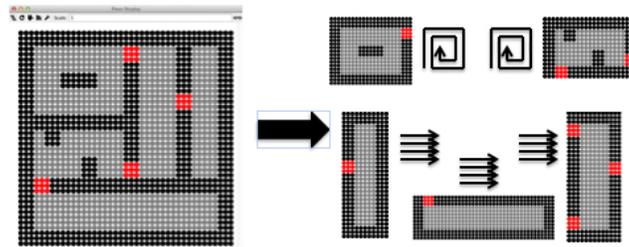


Figure 4: A central controller assigns stationary Sentry agents to strategic locations, displayed in red, to partition the search space into sub-regions. Trade-offs between the two search formations make partitioning the search region an optimization problem. Implemented in a DDDAS framework, the simulation supports dynamic formation reassignment when UAV failure impacts mission objectives (McCune and Madey 2013).

This agent-based simulation can be implemented within a DDDAS framework so real-time information can update the model, and, if necessary, aid the central controller in dynamic formation reassignment.

3.4 Communication

Another project investigated intra-swarm communication and effects on performance within a DDDAS framework (Purta et al. 2013). In this experiment, a UAV swarm is tasked with discovering all stationary targets within a 2-D environment. UAV swarm mobility was modeled after the Icosystem Swarm Game (Bonabeau 2002), a well-known swarm rule set where each agent associates non-mutually with one or two other agents in the swarm, designated as a helpers of a particular agent. For this implementation, UAVs are initialized as only either aggressors or defenders and position themselves according to their helpers, summarized in Figure 5. To control the swarm, the model was augmented with a leader-follower mechanism, where a designated leader agent is selected as the first helper agent of all other agents. The leader agent visits predefined waypoints in order for the swarm to search the entire environment.

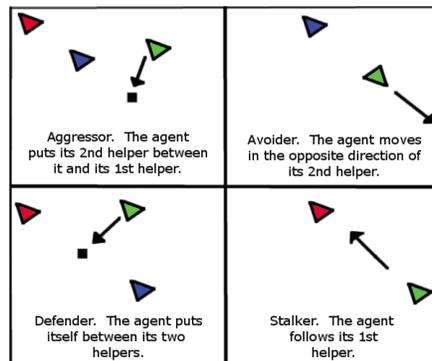


Figure 5: Agent mobility rules of the IcoSystem Swarm Game. Aggressor and Defender agents were implemented with the addition of a single Leader agent. The Leader is assigned as the first helper of every other agent, and traverses predefined waypoints throughout the environment to ensure quality coverage (Purta et al. 2013).

For efficient searching, UAVs should be positioned far enough away from one another so that search sensors do not overlap. However, for proper mobility, each agent must maintain a communication link with two helpers in order to correctly reposition itself. The baseline implementation required all agents to be within a given communication radius of one another, at the expense of overlapping search coverage and increased risk of collision. Another implementation supported multi-hop communication, permitting agents to be

farther away from helpers. Experiments were conducted with the baseline and multi-hop implementations to evaluate performance in terms of number of collisions and number of targets discovered. For the baseline experiments, agents were first positioned randomly about the environment, then later were modified to be evenly distributed around a circle in the center of the environment, with the search radius increased for each agent. Experiments with the multi-hop communication swarm had the same waypoints, initial positions, and sensor parameters as the second set of baseline tests, with communication routes determined by the path costing the least amount of energy.

The architecture of the experimental test-bed employs the DDDAS paradigm. A multi-agent simulation, programmed in Java utilizing the MASON multi-agent simulation toolkit, forms the foundation of the experiments. Simulation input and output is facilitated by RESTful web services running locally on an Apache Tomcat server. Web services communicate between the simulation and a separate user console, capable of displaying real-time UAV data. Real-time data may be any real-world representation of a UAV, whether physical drones or a separate simulation, implemented for proof-of-concept. The architecture is depicted in Figure 6.

The mixed results of the experiments support the need for a DDDAS framework. While multi-hop communication was hypothesized to increase the number of targets discovered, results showed that fewer targets were found on average than in baseline tests with non-random initial placement. However, since multi-hop configuration minimized total energy used by the swarm to communicate, some situations may still find such a design beneficial. Other performance trade-offs were identified, for further discussion of results, refer to (Purta et al. 2013). Intra-swarm communication was demonstrated to effect swarm performance in complex ways an operator may not immediately able identify or understand. The DDDAS implementation of this test-bed allows for a simulation to be executed with real-time data in faster than real-time. From the console, an operator is able to seamlessly adjust mission parameters based on simulation output, giving the operator an effective support tool for complex missions.

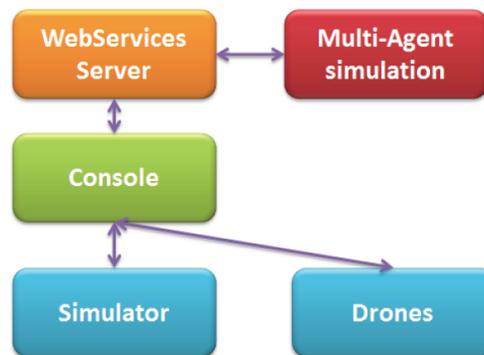


Figure 6: The experimental test-bed incorporating DDDAS architecture. Experiments utilized an agent-based simulation developed in Java using the MASON toolkit. Web services support both the real-life representations of UAVs driving simulation input, as well as the simulation output dynamically steering the real-time measurements of the UAVs (Purta et al. 2013).

3.5 Swarm Control Through BOIDS

Lastly, a fourth project explores UAV swarm command and control with BOIDS flocking behavior (Madey and Madey 2013). One of the earliest examples of swarming behavior (Reynolds 1999), BOIDS models the emergent flocking formation of birds through with three local rules that govern an agent flight vectors:

- Alignment- agents align their heading with their neighbors
- Cohesion- agents steer towards a group of neighbors

- Separation- agents steer away from their neighbors if they get too close

Subsequent rules were added that refine the flight vector by considering both current and predicted locations of specific neighbors (Reynolds 1999, Slear 2006). By manipulating the influence of each rule on the flight vector, the swarm can be controlled. Two surveillance scenarios were developed to evaluate the effectiveness of BOIDS for UAV swarm control.

3.5.1 Plume Coverage

The first scenario models the surveillance and tracking of a contaminant plume, building upon previous work in the domain (Daniel and Wietfeld 2011, Kovacina et al. 2002). In a 2D space, the contaminant plume is represented as a circle with the highest contaminant concentration in the center and dissipating outward towards the edges. The objective of the UAV swarm is to provide the controller with an accurate map of the plume and the distribution of the contaminant. Each UAV is capable of measuring the concentration from its current location, and broadcasting measurements to nearby neighbors, within a radius controlled by the swarm operator. Agents follow the cohere rule and navigate towards the neighbor with the highest reading, with a steering radius also controlled by the operator. The simulation displays the standard deviation of sensor measurements in real-time, providing the operator with statistical information to support control decisions. An example of utilizing swarm control to achieve mission objectives is depicted in Figure 7.

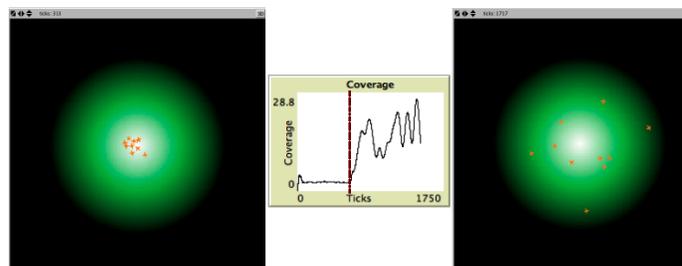


Figure 7: Illustrative example of swarm command and control with the BOIDS flocking rule set for a contaminant plume surveillance mission. The left-most image shows UAVs gathering around the highly concentrated center of the plume, controlled with a tight turning radius. In the center, real-time statistics reveal that the UAVs are not adequately dispersed about the plume, so the controller reduces the agent turning radius to cover more of the plume and obtain a greater deviation in measurements, seen right (Madey and Madey 2013).

3.5.2 Target Tracking

The second scenario implements two types of flocking agents, Searchers and Pursuers, to model the search and tracking of mobile targets. Searcher agents navigate through the search space according to the original three BOIDS rules, with the objective to maximize coverage. When a target is detected, a Searcher communicates the location to a Pursuer agent, which implements BOIDS seeking behavior to intercept the target at a predicted position. The ratio of Searchers to Pursuers is controlled by the operator and can be varied based on performance metrics, including total detections, successful pursuits, and the number of identifications for each unique target. Similar to the previous scenario depicted in Figure 7, real-time statistics support swarm control decisions of the operator. Figure 8 illustrates the target-tracking simulation.

4 CONCLUSIONS AND FUTURE WORK

UAVs will continue to play an important role as technology advances. Four projects were presented that explore swarm control of UAVs. While swarms are inherently decentralized, UAV swarms in some



Figure 8: Screen capture of the target tracking scenario. Gray arrowheads represent target vessels in a river, green airplanes represent Searcher UAVs, orange airplanes represent unassigned Pursuers, while red airplanes represent Pursuers actively tracking (Madey and Madey 2013).

manner must be centrally controlled to allow for a ground operator. Each project explored different aspects of swarm control implemented through an agent-based simulation and applied to the DDDAS framework. The mission scheduling project introduced an agent-based control framework for assigning missions based on minimum estimated cost. Through cooperative search, we demonstrated how swarm formations can be centrally assigned before operating autonomously. A third project demonstrated the impact of communication protocol on swarm behavior. Lastly, the swarm control scenarios demonstrated mission objectives achievable through BOIDS flocking behavior.

Many areas for potential improvement exist. For mission scheduling, improved realism can be obtained by accounting for non-static speed and collision avoidance. For cooperative search formations, an expression of parameters can be formulated in order to automatically optimize formation assignments. For communication, sorting of waypoints and found targets can be implemented minimize total distance traveled by the swarm. Finally, realism can be improved in the contaminant plume scenario by incorporating weather factors.

REFERENCES

- Altshuler, Y., A. M. Bruckstein, and I. A. Wagner. 2005. "Swarm robotics for a dynamic cleaning problem". In *IEEE Swarm Intelligence Symposium*, 209–216.
- Altshuler, Y., and V. Yanovski. 2005. "Dynamic Cooperative Cleaners Various Remarks". Technical report, Technical report.
- Altshuler, Y., V. Yanovsky, I. A. Wagner, and A. M. Bruckstein. 2005. "The cooperative hunters-efficient cooperative search for smart targets using uav swarms". In *Second International Conference on Informatics in Control, Automation and Robotics (ICINCO), the First International Workshop on Multi-Agent Robotic Systems (MARS)*, 165–170.
- Axelrod, R. 2006. "Advancing the art of simulation in the social sciences". In *Simulating Social Phenomena, volume 456 of LNEMS*. Springer, 1997. João Balsa, Luis Antunes, Ana Respício, and Helder Coelho. *Autonomous inspectors in tax compliance simulation*. In *Proc. 18th European Meeting on Cybernetics and Systems Research*. Citeseer.
- Bonabeau, E. 2002. "Agent-based modeling: Methods and techniques for simulating human systems". *Proceedings of the National Academy of Sciences of the United States of America* 99 (Suppl 3): 7280–7287.

- Bonabeau, E., M. Dorigo, and G. Theraulaz. 1999. *Swarm intelligence: from natural to artificial systems*, Volume 4. Oxford university press New York.
- Cambone, S. A. 2005. *Unmanned aircraft systems roadmap 2005-2030*. Defense Technical Information Center.
- Daniel, K., and C. Wietfeld. 2011. "Using Public Network Infrastructures for UAV Remote Sensing in Civilian Security Operations". Technical report, DTIC Document.
- Darema, F. 2004. "Dynamic data driven applications systems: A new paradigm for application simulations and measurements". In *Computational Science-ICCS 2004*, 662–669. Springer.
- Garcia, R., and L. Barnes. 2009. "Multi-UAV simulator utilizing X-Plane". In *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA*, 393–406. Springer.
- Gaudiano, P., E. Bonabeau, and B. Shargel. 2005. "Evolving behaviors for a swarm of unmanned air vehicles". In *Swarm Intelligence Symposium*, 317–324. IEEE.
- Gaudiano, P., B. Shargel, E. Bonabeau, and B. Clough. 2003. "Control of UAV swarms: What the bugs can teach us". In *Proceedings of the 2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations Aerospace Conference, San Diego, CA*, 6624–6624.
- Hodson, H. 2013. "Hand-delivered parcels find their way to you via the crowd". *New Scientist* 218 (2917): 17–18.
- Khaleghi, A. M., D. Xu, Z. Wang, M. Li, A. Lobos, J. Liu, and Y.-J. Son. 2013. "A DDDAMS-based Planning and Control Framework for Surveillance and Crowd Control via UAVs and UGVs". *Expert Systems with Applications*.
- Kovacina, M. A., D. Palmer, G. Yang, and R. Vaidyanathan. 2002. "Multi-agent control algorithms for chemical cloud detection and mapping using unmanned air vehicles". In *Intelligent Robots and Systems. IEEE/RSJ International Conference on*, Volume 3, 2782–2788. IEEE.
- Lamont, G. B., J. N. Slear, and K. Melendez. 2007. "UAV swarm mission planning and routing using multi-objective evolutionary algorithms". In *Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on*, 10–20. IEEE.
- Luke, S., C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. 2005. "Mason: A multiagent simulation environment". *Simulation* 81 (7): 517–527.
- Madey, A. G., and G. R. Madey. 2013. "Design and evaluation of UAV swarm command and control strategies". In *Proceedings of the Agent-Directed Simulation Symposium*, 48–55. Society for Computer Simulation International.
- Madey, G. R., M. B. Blake, C. Poellabauer, H. Lu, R. R. McCune, and Y. Wei. 2012. "Applying DDDAS Principles to Command, Control and Mission Planning for UAV Swarms". *Procedia Computer Science* 9:1177–1186.
- McCune, R. R., and G. R. Madey. 2013. "Agent-based simulation of cooperative hunting with UAVs". In *Proceedings of the Agent-Directed Simulation Symposium*, 56–61. Society for Computer Simulation International.
- Purta, R., S. Nagrecha, and G. Madey. 2013. "Multi-hop Communications in a Swarm of UAVs". In *Proceedings of the Agent-Directed Simulation Symposium*, 32–39. Society for Computer Simulation International.
- Rasmussen, S. J., J. W. Mitchell, P. R. Chandler, C. J. Schumacher, and A. L. Smith. 2005. "Introduction to the MultiUAV2 simulation and its application to cooperative control research". In *American Control Conference*, 4490–4501. IEEE.
- Reynolds, C. W. 1999. "Steering behaviors for autonomous characters". In *Game Developers Conference Proceedings*, 763–782.
- Russell, M. A., G. B. Lamont, and K. Melendez. 2005. "On using SPEEDES as a platform for a parallel swarm simulation". In *Proceedings of the 37th conference on Winter simulation*, 1129–1137. Winter Simulation Conference.

- Slear, J. N. 2006. "AFIT UAV swarm mission planning and simulation system". Technical report, DTIC Document.
- Varela, G., P. Caamamo, F. Orjales, A. Deibe, F. López-Peña, and R. J. Duro. 2011. "Swarm intelligence based approach for real time UAV team coordination in search operations". In *Nature and Biologically Inspired Computing (NaBIC), Third World Congress on*, 365–370. IEEE.
- Wei, Y., M. B. Blake, and G. R. Madey. 2013. "An Operation-Time Simulation Framework for UAV Swarm Configuration and Mission Planning". *Procedia Computer Science* 18:1949–1958.
- Wei, Y., G. R. Madey, and M. B. Blake. 2013. "Agent-based Simulation for UAV Swarm Mission Planning and Execution". In *Proceedings of the Agent-Directed Simulation Symposium*, 7–14. Society for Computer Simulation International.

ACKNOWLEDGMENTS

This research was supported in part under the Dynamic Data Driven Applications Systems (DDDAS) Program, and the National Science Foundation award #1063084. The authors would also like to thank the University of Notre Dame Center for Research Computing, and Christian Poellabauer.

AUTHOR BIOGRAPHIES

RYAN MCCUNE is a PhD candidate in Computer Science at the University of Notre Dame. His research interests include swarm intelligence, agent-based modeling, and multi-agent systems. He is a member of the ACM and IEEE Computer Society. His email is rmccune@nd.edu and his web page is <http://www.nd.edu/~rmccune>.

YI WEI is a Software Development Engineer at Microsoft. He received his PhD from University of Notre Dame. His research interests are service-oriented computing, cloud computing, and agent-based modeling and simulation. He is a member of IEEE and ACM. His email is weiyi.nd@gmail.com and web page is <http://www.nd.edu/~ywei1>.

RACHAEL PURTA is a PhD candidate in Computer Science at the University of Notre Dame. Her research interests include mobile sensor networks, security, and agent-based modeling. Her email address is rpurta@nd.edu and web page is <http://www.nd.edu/~rpurta>.

ALEXANDER MADEY is a senior at Trinity High School in South Bend, IN. His interests include agent-based modeling and simulation. His email address is alexmadey@gmail.com.

M. BRIAN BLAKE is an Affiliate Professor of Computer Science at the University of Notre Dame, as well as Dean of the Graduate School and Vice Provost for Academic Affairs at the University of Miami. He conducts applied research in the areas of service-oriented computing, software engineering, and enterprise integration. His email address is m.brian.blake@miami.edu and web page is <http://www.cs.miami.edu/~blake>.

GREGORY MADEY is a Research Professor in the Department of Computer Science and Engineering at the University of Notre Dame. His research includes topics in bioinformatics, emergency management modeling and simulation, agent-based modeling and simulation, web portals for scientific collaboration, and scientific databases. He is a member of the ACM, AIS, IEEE Computer Society, INFORMS, and the Society for Computer Simulation. His email address is gmadey@nd.edu and web page is <http://www.nd.edu/~gmadey>.