

DETERMINING ROBUST CONTROL METHODS FOR COORDINATED UAVS IN VARYING MISSION ENVIRONMENTS

Jacquelyn Banas
Anthony Gray, Gaël Goron, Wil Roberts, Somil Shah, Paola Zanella
Dr. Kelly Griendling, Dr. Dimitri Mavris

Georgia Institute of Technology
Atlanta, Georgia
United States of America

ABSTRACT

Recent advances in technology have allowed for smaller, cheaper, and more versatile unmanned aerial vehicles (UAVs) to become prolific throughout both the military and civilian spheres. Using many small, VTOL capable UAVs in a multi-agent system (MAS) allows for continuous tracking of multiple targets, surveillance below tree-lines, and closer views of the targets. MAS coordination schemes and enhanced vehicle autonomy have become increasingly important to help alleviate operator workload when using a group of such UAVs in a hostile environment. Current aerospace applications of coordinated MAS's typically rely on leader-follower control schemes, which are inherently vulnerable to communication and sensor loss and often require human input for complex tasks. Meanwhile, other industries have successfully implemented decentralized methods for coordinating numerous vehicles, requiring only local information about neighboring agents in order to determine new positions. The decentralized methods do not require communication or GPS functionality, ensuring a more robust MAS in contested environments. This project, called "UV-CoRE," developed a simulation framework to characterize the benefits and limitations of various MAS control methods. The UV-CoRE framework was based on an existing, in-house JAVA simulation environment, adding multi-agent coordination capabilities, various 2-D mission scenarios, and a variety of MAS coordination schemes for comparison. Results from the simulation have shown that a blending of decentralized and leader-based MAS control methods is necessary to design a safe, effective, and reliable coordinated multi-agent system. The UV-CoRE simulation environment is capable of modeling custom agents, control algorithms, and mission scenarios for related future studies.

NOMENCLATURE

2-D	Two-dimensional
AI	Artificial Intelligence
ASDL	Aerospace Systems Design
DoE	Design of Experiments
GPS	Global Positioning System
IR	Infrared Radiation
ISR	Intelligence, Surveillance, Reconnaissance
LED	Light-Emitting Diode
LIDAR	Light Detection and Ranging
MAS	Multi-Agent System
NATO	North Atlantic Treaty Organization
NPS	Naval Postgraduate School (U.S)
RRT	Rapidly Exploring Random Tree
SONAR	Sound Navigation and Ranging

UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UV-CoRE	Unmanned Collaboration & Research
VTOL	Vertical Take-off and Landing

1. INTRODUCTION

In the last decade, military operations in areas such as Afghanistan and Libya have added significant numbers of unmanned aerial vehicles (UAVs) to NATO allies' arsenals¹. These unmanned agents have become crucial elements in the global fight against extremism as well as humanitarian missions: they allow for targeted intelligence operations in areas where it might be dangerous, costly, or time-consuming to send human troops and manned aircraft. Increased intelligence helps prevent civilian casualties and keeps commanders better informed in rapidly-changing, complex combat environments, supporting NATO's vision¹ of "360 degree situational awareness." The defense industry has emphasized the need for better continuous tracking of targets,

especially in urban environments where obstacles and areas of low visibility hinder current ISR capabilities.

Recently, advances in computer hardware, software, and data processing have allowed for smaller, cheaper, and more versatile UAVs to become available both in military and civilian markets. New commercial, off-the-shelf (COTS) solutions for security and surveillance offer a variety of options to fill gaps in intelligence-gathering operations. The Aeryon SkyRanger™ sUAS multi-copter, for example, can handle high winds, maneuver between buildings, and go below tree lines while recording high-resolution EO/IR video for up to 50 minutes of continuous operation—for an almost negligible acquisition cost compared to today's unmanned assets^{2,3,19}.

A coordinated multi-agent system (MAS) comprised of several of these new, inexpensive UAVs would add immensely to the ISR capability that regular ground infantry troops currently have, especially without the levels of intelligence support that are typically only available to more specialized operations. Also, for special operations, a coordinated MAS could effectively replace multiple levels of expensive intelligence-gathering aircraft. Depending on sensor and bandwidth availability, future systems could relay pictures, video, sound, and IR signatures directly to ground troops as well as to central command centers. A notional illustration of this scenario is shown in the Operational View-1 (OV-1) of Figure 1. A coordinated MAS of UAVs such as the one shown in the OV-1 would greatly reduce mission costs, increase the quality of intelligence, and reduce the time delay between intelligence gathering and troop action.

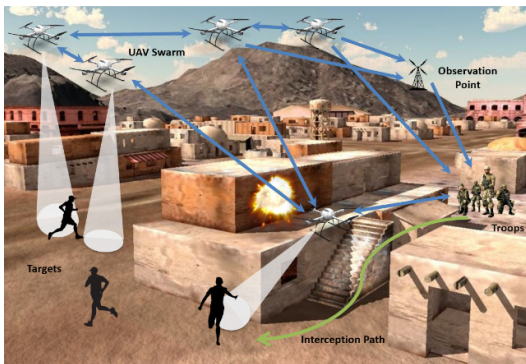


Figure 1. Operational View-1 (OV-1) for a Notional Urban Infantry Mission Using an MAS of UAVs.

2. MOTIVATION

Conducting flight operations and monitoring data from multiple aircraft simultaneously could easily overwhelm operators in the field. Therefore multi-agent coordination schemes and vehicle autonomy have become increasingly important to alleviate operator workload and allow for increased focus on the sensor data instead. As UAVs mature in terms of autonomy and coordination, multi-agent systems will become increasingly effective in more complicated missions.

Currently, most aerospace applications of coordinated multi-agent systems rely on leader-follower control schemes and require human input for complex tasks. This control design passes information quickly and reliably when all agents are in communication, but it is inherently vulnerable to communication and sensor dropouts.

For example, the U.S. Naval Postgraduate School (NPS) has successfully coordinated up to 50 aircraft in its ARSENL project⁶. However, the aircraft were reliant on constant Wi-Fi communication, had two designated subgroup “leaders”, and were remotely-controlled by two human operators. The U.S. Office of Naval Research (ONR) worked instead with fully-autonomous aircraft, but they again used a leader-follower scheme and only managed up to 9 aircraft at a time⁸. Loss of a “leader” aircraft in either case could result in mission failure or possibly loss of the entire group of aircraft.



Figure 2. NPS ARSENL Program Controlled up to 50 Aircraft in Two Sub-Groups.⁶

Meanwhile, non-aerospace applications have successfully implemented decentralized methods for rather complex tasks, requiring only local information about neighboring agents in order to reposition vehicles. For example, Harvard University's “Kilobots” project successfully managed a group of 1,024 small, inexpensive, fully-autonomous ground vehicles¹⁰. The group was able to self-assemble into various complex shapes when given high-level human requests, relying only on a local data exchange between nearby infrared LED transmitters and infrared photodiode receivers.

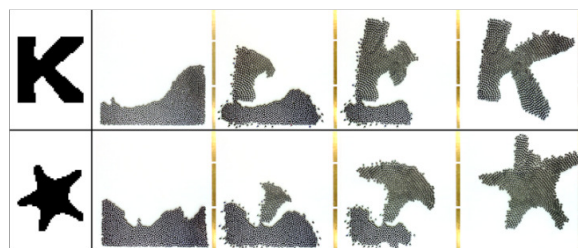


Figure 3. Harvard Kilobots Form Shapes Using only Local Sensing and Communication²⁰.

The decentralized methods often take much more time to move agents to desired locations, require very sterile and well-understood environments, and may be

unpredictable after long periods of time. However, focusing primarily on using local information and allowing agents to self-determine their positions allows for a more emergent system, which will be safer and more robust in contested environments.

Further research is necessary to determine the proper blending of available control and coordination methods in order to design a safe, effective, and reliable coordinated multi-agent system.

3. LITERATURE REVIEW

The general literature search for this project spanned a multitude of subjects in order to create a realistic mission scenario and incorporate state-of-the-art MAS control methodologies. Research started with general information about UAVs as well as military, police, and rescue groups' operations with current manned and unmanned assets. Next, detailed information was gathered on a variety of available control methods to coordinate and most effectively use a group of UAVs to complete missions with differing metrics and constraints.

3.1 Market Basis for Multi-Agent Systems

Research has shown that UAVs offer great cost and capability benefits, in addition to keeping humans out of dangerous situations. As shown in Figure 4, unmanned aircraft such as the MQ-1C Grey Eagle, MQ-9 Reaper, or MQ-1 Predator can offer up to three times the endurance of the comparable manned MC-12 Liberty⁷, enabling longer missions that span wider distances. Manned aircraft such as the Liberty that are used for mid-altitude surveillance often cost 10's of millions of dollars for acquisition alone, while the electric quadcopter replacements that are envisioned in this project are on the order of \$2,000-\$10,000² each. Furthermore, the small, electric quadcopters require comparatively negligible maintenance and operating costs per mission.

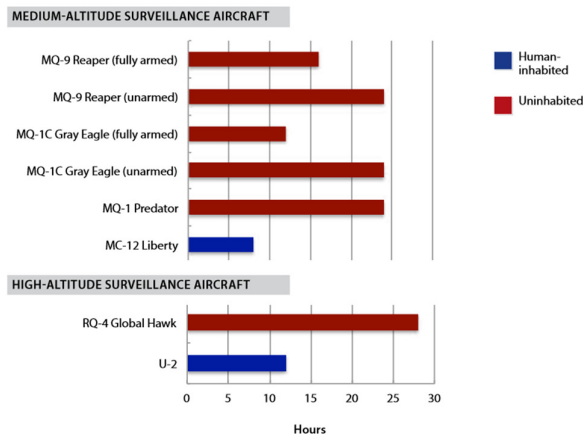


Figure 4. Surveillance Aircraft Endurance⁷.

Urban infantry special operations missions are often complex events, involving a variety of assets and intermediaries to aid in intelligence gathering and mission execution: mid-altitude MQ-1 Predator-type aircraft support electronic warfare (EW), intelligence

surveillance reconnaissance (ISR), and other fixed-wing as well as rotary assets below. Ground troops may also carry hand- or rail-launch UAVs such as the AeroVironment RQ-11 Raven to collect additional information closer to their locations. Intelligence gathered by the aircraft above goes through central command groups before reaching the troops, adding additional delays into the system and not always providing the ground troops with current information.

With smaller UAVs added into the troop's arsenal and launched from a safe starting "observation point," the platoon-level soldiers could easily view intelligence data real-time and help commanders make more informed decisions. In some cases, the troops could remain at the observation point gathering data and would never need to enter the hostile area.

3.2 Multi-Agent System Control

Establishing mature methods for coordination and control methods for the multi-agent system is key to effectively using the assets without overwhelming the troop's resources or actually introducing additional risk into the missions. In order to conduct a mission successfully with minimal human operator involvement, the agents as well as the MAS need proper path planning, obstacle avoiding, target detection, and agent coordination logic.

In controlling such a multi-agent system, sometimes called a "swarm," the robotics industry offers some textbook criteria for developing safe and robust control methods. The "Four Laws of Swarm Control" recommend the following: proper swarm methods must be local, scalable, safe & reactive, and emergent⁴. "Local" limits the agents to acting on information that they have sensed or know themselves; "scalable" refers to a system that can handle many agents or few and also remain functional with larger or smaller processing & memory capabilities. "Safe & reactive" requires that each agent is capable of basic autonomous vehicle operation, with a reactive artificial intelligence (AI) capability that properly responds to changing environments and does not require input from another agent or system. Finally, an "emergent" system is one where the local AI rules transfer to the global system without creating problems for the overall group, and the agent behavior is predictable. The system should be designed such that an operator can control the agents from a "swarm" level, without having to interact directly with any of the individual agents in the MAS.

3.2.1 Swarm Coordination Methods

Numerous methods of coordinating a group of UAVs were found during the literature search. The most applicable for this project were centralized techniques using leader-follower relationships and decentralized techniques using consensus, partitioning, and distributed networking control methods.¹⁸

As previously discussed, the leader-follower scheme is the most common used today for aerospace applications. This method, illustrated in Figure 5,

designates a leader that either gives explicit instructions to each of the agents or at a minimum provides a guiding direction for the others to follow. This method is very efficient and predictable; it can handle complicated environments better than the other methods if the leader has proper instruction, sensing capabilities, and communication with the followers. If necessary, a human can easily override and control the system to finish a mission.

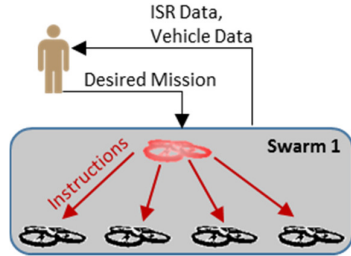


Figure 5. Leader-Follower Method for MAS Control.

For even more efficient movement or to track multiple targets, a hierarchical structure with subgroups led by subleaders could also be implemented—similar to what was employed in the NPS ARSENL project and shown in Figure 6. However, these systems are inherently susceptible to problems that arise often in contested environments: communications and/or sensor dropouts as well as agent loss would leave agents without proper direction, in the absence of other coordination methods.

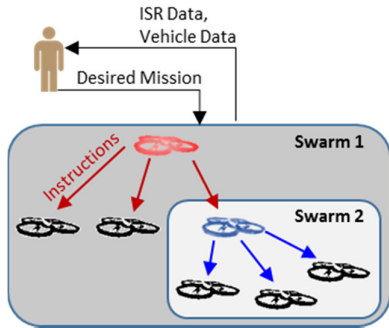


Figure 6. Leader-Follower with Subgroups Method for Large MAS Control and Multiple Targets.

On the decentralized side of the spectrum, the simplest coordination technique is using a “boids” flocking method, pictured in Figure 7. Agents continuously look at near-neighbors based on their own near-field sensing capabilities to judge relative positions and heading. A pre-programmed desired offset from other agents allows each to maintain safe separation as well as proper cohesion to the moving group. Each agent seeks to maintain this offset and also align with the heading of its neighbors by adjusting its velocity and heading, as necessary. This simple technique can be quite effective at moving a group together in a complex environment, but it does not allow for each agent to do a specific, different task.

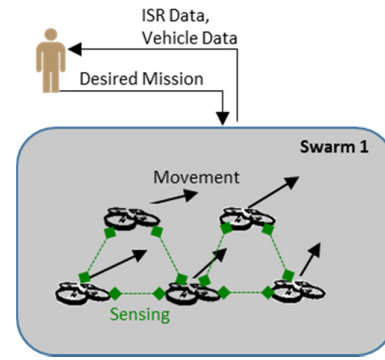


Figure 7. Boids Method for MAS Control.

A more complicated, decentralized control scheme using distributed networking methods, via weighted rendezvous equations, allows for the mathematically-provable assignment of agents to relative coordinates in space. This method requires only local, relative position information that could easily come from very small, inexpensive infrared (IR) or SONAR sensors. Agents look at each near neighbor with their own sensors and move based on relative locations as compared to a pre-programmed, desired position or set of positions. “Weighted rendezvous” refers to the modified consensus equation used for this method⁴

$$(1) \quad \dot{x}_i = -\sum_{j \in N_i} w_{i,j} (\|x_j - x_i\|) (x_j - x_i)$$

where $\dot{x}_i$ is the state vector of agent i , N_i is the near-neighbor set of agent i at a given time, $w_{i,j}$ is the “weighting” to alter the exact desired position of agents, and j refers to each of the other agents²¹. The parameters x_j and x_i refer to the position of agent i and j , respectively. Weights are modified to prevent collision among the agents and also to place them in specific, desired locations.

Using this method, individual UAVs could form patterns or move from one point to another without directly communicating with each other or carrying expensive, heavy, and often unreliable LIDAR, GPS and Wi-Fi equipment. Any agent could be lost without disturbing the group’s ability to perform the actions prescribed. However, this method requires some hard-coded instructions or another source of carefully-designed AI to generate the “desired” locations and move the entire group towards a target. The agents may also take a long time to reach the desired final positions, and near-neighbor sensor information may be lost if the agents get separated for even a short duration of time. Without other levels of programming added to the code, there is no guarantee that all the agents will make it to a desired point in a changing environment with obstacles, and it may take a very long time for them to get there even in a controlled situation.

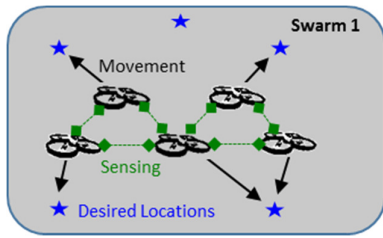


Figure 8. Decentralized MAS Control Using a Weighted Rendezvous Method.

Another option for decentralized control of the UAVs would be a partition-based approach: in this method, the search area is divided in one of many ways, and sectors are assigned to each agent or groups of UAVs. This assignment could be decided individually by each agent, by a human operator, or by consensus and communication among the agents in a MAS. A notional illustration of a sector search based on consensus assignment is shown in Figure 9.

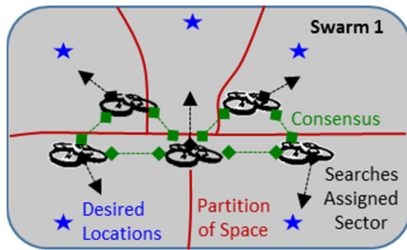


Figure 9. Partitioning and Consensus Method for MAS.

Some common partitioning methods are based on comparisons of the Euclidian distance of the agent to the intended position or a velocity-based “distance,” which can help increase the overall efficiency of the group by choosing the agent that will reach the target first¹¹. The main drawback of this method is that the area needs to be split up into sectors first, which requires sound knowledge about the contents of that area or good communication and sensing capabilities within the MAS. This method is most successful when the environment is well-understood and very stable over time.

Finally, a decentralized consensus method without partitioning could also be considered. This requires voting between the agents to determine individual movement in a more general sense, and so this method relies heavily on advanced pre-programmed AI as well as good communication and location information. The “next” position for each agent or the group of agents takes additional time to determine compared to the leader-follower scheme, but it may be significantly faster than the weighted rendezvous decentralized method when communication is good and AI algorithms are well-tuned. In a realistic urban scenario, this method breaks down quickly: agents that cannot communicate with each other, get lost easily, and the group may not “agree” on a next position correctly if location information is faulty for even one of the agents.

3.2.2. Path Planning Methods

Several methods for individual agent and coordinated group path planning are available in the general body of research, but only a few are applicable for vehicles flying in a complex, changing 3-D world. An MAS used in an urban mission needs a method that can handle the complexity of the situation while also quickly and reliably moving the agents to the target location(s). Data processing constraints must also be considered, as these agents often are limited by their hardware.

“Maximum Principle” methods attempt to minimize “cost” but were found to be difficult to implement, and they do not provide feedback. Decomposition-based methods discretize the search area, similar to the partition-based coordination technique discussed previously. Dynamics are ignored and desired movement is represented by waypoints, so this method cannot handle high dimensionality either and thus is not well-suited for VTOL-capable UAVs in urban scenarios.

“Dynamic programming” uses partial differential equations and yields accurate and fast results, but these methods often cannot handle high dimensionality. One notable method that is a subset of the dynamic programming methods is the Gradient Descent control scheme. This method, based on a first-order optimization technique, generates a field of repulsion and attraction vectors throughout a known map. Agents are simply repelled from obstacles and other agents while being attracted to targets at differing, adjustable gains. This method requires prior knowledge of the search area and does not necessarily perform well with multiple targets, but it is often successful in simple robotics simulations.

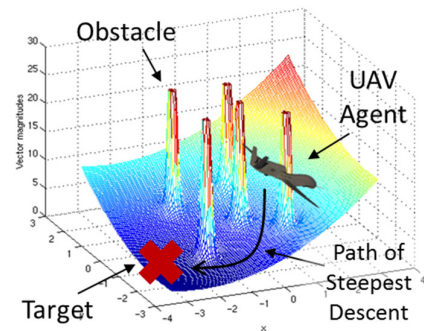


Figure 10. Gradient Descent Path Planning Method.

“Sampling methods” can handle high dimensionality by calculating decompositions in real-time and generating a path by sampling points within sectors. These methods are relatively immature and are still being developed to increase robustness, optimality, and efficiency. However, if properly implemented, sampling methods could provide the reactive path planning that an MAS requires in complex, changing urban environments.

Two notable sampling methods are A* and RRT*. A* is a very commonly-implemented method that generates an optimal path based on a comparison of

available paths. This method performs well in situations where clear maps and obstacle information are available, but it is not as successful in dynamic environments like cities. Rapidly exploring random tree (RRT) methods build a path tree one branch/node at a time by sampling randomly within the search space for the next branch and node. This could be combined with sensor data to check for unobstructed paths in real-time, if enough processing capability is available. The RRT* variation follows the same approach but looks two nodes ahead on each branch to determine the local optimal path. RRT* can be modified to search a particular sector or region to guide the movement towards a known target location more efficiently, and it tends to generate space-filling trees that expand outward, which results in overall more success in reaching a goal in an environment filled with many obstacles.

3.2.3. Swarm Formations and Search Patterns

For some missions, specific formations and search patterns are useful to more efficiently locate and track a target. Formations such as a V-shape (similar to that of flocking birds) helps to reduce fuel burn for fixed-wing aircraft traveling to search areas far from their launch base. Circles, stars, or pyramid shapes around a datum allow hover-capable UAVs to surround a target and return continuous video footage from multiple angles. Lines and offset variations allow the UAVs to move through tight spaces, to minimize visible agents in a hostile environment, or to maximize sensor coverage through a wide expanse of space.

Search patterns used by the U.S. Coast Guard⁵ were also investigated for use in this project, and a subset are shown in Figure 2. The simplest method, flying a track line, follows the expected path of the target. Flying a parallel track allows for uniform coverage over a large search area, when an approximate starting target position is known. A creeping line method adds some complexity to the track line path for additional coverage of a narrow but long search area. Sector searches are most useful when the target location is known within a small area, and expanding squares are best when both the target location is known within a small area and a concentrated search is desired.

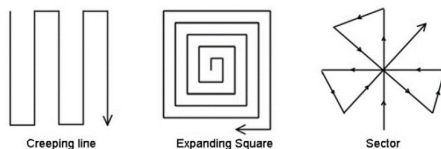


Figure 2. U.S. Coast Guard Search Patterns⁵.

4. TECHNICAL APPROACH

This project evaluated the effectiveness and robustness of various combinations of multi-agent control algorithms in a variety of mission scenarios. Using the parametric, 3rd person JAVA simulation environment called “UV-CoRE,” developed at Georgia Tech’s Aerospace Systems Design Lab (ASDL), numerous control schemes were modeled and

analyzed with respect to desired mission outcomes. An example mission was chosen to test the limits of each control method, and the number of agents, number of targets, size of search areas as well as target behavior and other factors were varied to find the critical design points for each control scheme. Metrics results were calculated from each simulation and compared to gain a better understanding each control method’s capabilities.

5. UV-CORE PARAMETRIC ENVIRONMENT

The tool used for this project is a Java-based modeling and simulation environment called “UV-CoRE.” Figure 11 shows the graphical interface when a simulation is started. A larger picture is included at the end of this paper, in Figure 20. Users can alter a variety of mission and vehicle properties by clicking on the buttons on the left hand side, and these parameters (as well as many others) can also be directly changed in the JAVA code. An example of customization options that are available is shown in Figure 12. Vehicle attributes set by the user are used to make rough estimates of drag, thrust required, and fuel burn at each time step of the simulation.

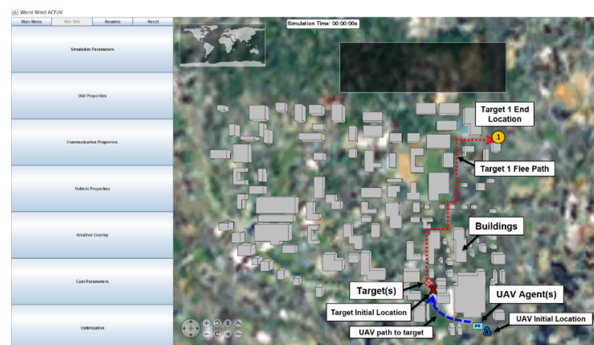


Figure 11. UV-CoRE Urban Simulation.

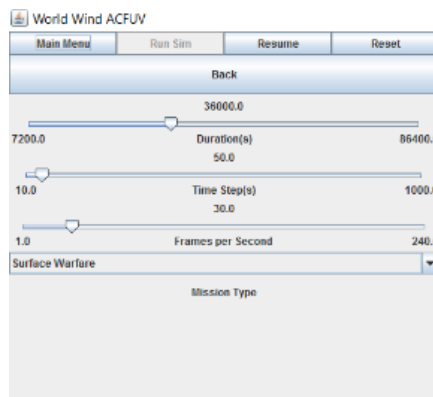


Figure 12. UV-CoRE User-Defined Properties.

The NASA World Wind package serves as the basis for location visualizations and agent symbology²², allowing for realistic landscapes and customization of agent positions, icons, and paths. While UV-CoRE had previously been developed at ASDL and used in prior projects, it had to be modified quite heavily to allow for multi-agent simulation capabilities, customized path planning, and to include obstacles.

5.1. UV-CoRE Simplifications & Assumptions

Agents and targets are modeled as point masses and represented by standard military icons²². Starting location is set by the user by providing the desired latitude and longitude. Analyses for this research project will be limited to 2-D experiments, but the environment also has the capability to do 3-D simulations in the future, if desired. Figure 4 zooms in to show a detailed view of the target and multi-agent group in this particular urban mission scenario. Agents in the “swarm” were assumed to be electric-powered, rotary UAVs, while the targets were modeled as human-like suspects.

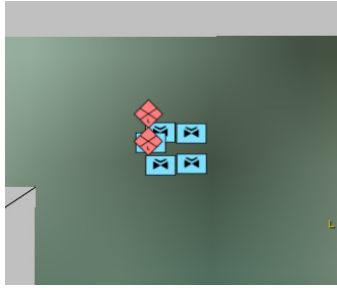


Figure 13. Multi-Agent System Tracking a Target.

To streamline the simulation and the coding process, and to isolate the effects of the MAS coordination schemes specifically, notable simplifications and assumptions were made in modeling other aspects such as communications, sensors, and vehicle performance. Simulation users can set both the “radar” and “radio” radiuses by entering numbers into the associated GUI fields. These correspond to an obstacle/agent detection radius and communication ranges, respectively. Initially, all messages from the sensors and radios were assumed to be 100% successful. Agents were also able to acquire perfect latitude and longitude information for themselves as well as all objects within their radar’s radius. In this way, the radar radius also sets the range of “near neighbors”. In addition to this, each agent’s electric power was assumed constant: it could not run out of battery and was capable of generating any thrust required. All of these assumptions were deemed valid given the relatively simple starting scenario and would need to be re-examined for different missions.

The parametric UV-CoRE environment is user-friendly, modular, and versatile enough to allow for the addition of more complex models for any of these parameters, if desired.

5.2. Multi-Agent System Capability

To allow for multi-agent system simulations, an additional group class called “swarm” was generated, and UAV agents adopt instances of that class as a group attribute. In some cases, such as in simulation with multiple targets, more than one swarm is generated during the simulation. Each “Swarm” class object stores a hashmap of all the agents within the group, and agents will register or de-register from each swarm hashmap as they enter or leave the

group. Two attributes of the swarm include a reference to a “Leader” agent as well as the desired MAS coordination method. The first agent to be generated in each swarm will be designated as a leader, whether the coordination method requires one or not. This attribute is simply not used in the cases where it is not necessary. When the user starts the simulation, the program loops through all of the swarms to call the desired AI of the proper agents within each group.

5.3. Simulated Example Mission

The primary mission used for analysis in this project was an “urban infantry assault” scenario. More precisely, a coordinated group of electric quadcopters was sent through a notional city full of obstacles to track a fleeing human target (or multiple targets). This example mission seeks to replicate an infantry-style action with a known starting target location. The MAS starts out at an observation point, proceeds towards the known initial target position, seeks to acquire the target with near-field sensors on each agent, and then pursues and tracks the target as it flees. The major elements of this scenario are depicted in Figure 14.

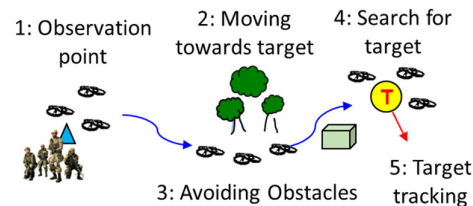


Figure 14. Primary Elements of Simulated Mission.

Quadcopters were chosen as agent models due to the relatively short mission duration, the high maneuverability needed to navigate through numerous obstacles (primarily buildings), and the portability of the aircraft.

A map of the Georgia Tech campus was simplified, digitized, and imported into the UV-CoRE framework to generate a two-dimensional “urban” environment, as shown previously in Figure 11. This simplified map of campus buildings provides varying shapes, sizes, and density of obstacles. For visualization purposes, the buildings are given a set “height,” but agents may not fly over the tops of these buildings—they are treated as infinitely-tall structures due to the 2-D nature of the simulation. Targets may run through the buildings, while pursuing quadcopter agents cannot see through or travel through them.

The target represents a person of interest in a crime. In this simulation, the crime is stealing the “T” off Tech Tower—unfortunately an infamous and common heist in Georgia Tech lore. Initially, only a single target with simple fleeing behavior was tested: this target initiates at the Tech Tower GPS location and begins fleeing at about 11.5mph once the quadcopter agents are within sight. The target then runs a prescribed, waypoint-based route through East Campus until reaching the bridge to Tech Square. Later, additional targets were

added to the simulation with more complex behavior to test the effectiveness of each control method when multiple and complex targets are present. A map of one set of target flee paths is shown in Figure 15.

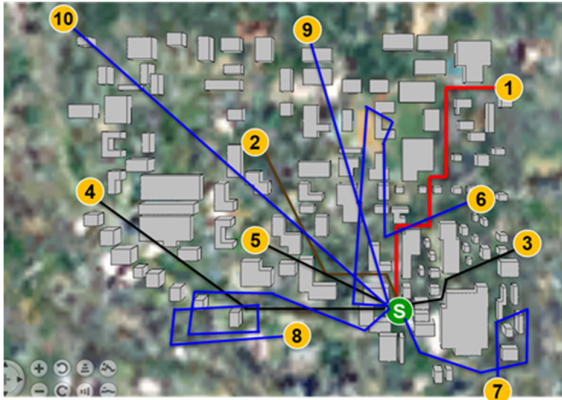


Figure 15. Example Set of Target Flee Paths.

Complex target behaviors also included different initial movement: when the simulation is started, the target may loiter in the nearby area or start running along its waypoint path. A summary of the initial target behaviors tested is outlined in Table 1.

Table 1. Example Target Initial Behaviors.

Number	Behavior Description
1	Wait until seen by UAV agents.
2	Loiter near initial point until seen by agents.
3	Flee immediately along waypoint path.

Pursuing quadcopter agents were modeled after existing commercially-available products. Maximum airspeed for these aircraft was set at 20kts to allow for quick maneuvering while also keeping up with the target. This value is really only important before the target is within sight, as the quadcopters are assumed to be generally faster than a human. Agent properties can be modified by the user as desired in UV-CoRE as shown in Figure 12, and additional custom agent objects can also be added into the simulation environment to test the sensitivity of the control methods to differing aircraft.

A simple communications model as well as a sensor model allow for instructions and sensor information to pass between the units. Currently, the sensor model is set to mimic near-field (infrared or SONAR) sensing capabilities that most small, autonomous aircraft contain. Additionally, unless turned off for a specific test, each agent also knows its GPS location. More advanced sensors such as LIDAR are not modeled here, since they are significantly more costly and less common among the smaller UAVs. However, additional models for advanced sensors could easily be implemented in future projects. This project seeks to determine robust control architectures in contested environments, so each method uses the least amount of information necessary from the sensors and communication to conduct each task.

Metrics tracking was also added to the simulation. For an urban mission, consistency of target tracking and time to reach the target are the two most important metrics, whereas cost is of a lesser concern. The time to reach the target is relatively straight-forward: this is calculated as the time for any of the agents to reach within a certain “sight” radius of the target. Once the agents reach the target, the amount of time that at least one agent has the target in sight is counted towards the “percentage of target tracking” metric. Robust combinations of control algorithms and agent formations for missions will be determined based on relative metrics results from the different simulation runs. Additional or different metrics could be included for other mission scenarios, as desired by simulation users.

5.4. Simulated MAS Control Methods

Given the time and resources available for this project, four primary MAS coordination method were compared, and one path planning/obstacle avoidance algorithm was implemented. Three of the coordination methods explored variations of the leader-follower scheme, while the fourth tested the boids method alone. The gradient descent method was used as the agents’ path planning and obstacle avoidance routine.

In all the leader-follower schemes, a similar general method for target tracking is implemented. At each time step, the leader of the swarm checks its radar and notes any obstacles or targets within its radar range. It then assigns the first target it sees to its swarm’s “target tracking” attribute. The leader then moves in the direction of the target, avoiding obstacles. If the leader does not see its target during a time step, it guesses the target’s next location by using the last known position and the last known heading. Anytime the leader sees its target, it marks the target as being seen at that time step. In the current control design, each leader stays with one target and will not mark other targets as “seen” nor will it follow new targets, even if they are in range of its radar. The simulation was programmed this way to maintain continuous surveillance of each target—a desire specifically voiced by a project sponsor—but the program could be altered easily to chase new nearby targets instead.

All the other agents that are not leaders are called “followers.” They check their radars for obstacles as well as leaders, other followers, and targets. If a target is seen within its radar radius, the follower also marks it as seen for that time-step, but this is not communicated back to the leader.

Depending on the coordination method implemented, leaders and followers will have different additional interactions. The following sections describe these methods in more detail. Experiments in the simulation focused on changing the mission scenario around these methods, to gain further understanding about the capabilities and limitations of each.

5.4.1 Leader/Follower with Boids

A baseline MAS control architecture was created using a passive leader-follower coordination scheme. A single leader agent (the first agent generated, in this case) used the gradient descent method to move towards the target and avoid obstacles, while the followers (all other agents) used a boids method for coordination and collision avoidance. The boids method, as described previously, ensures separation between agents, maintains cohesion among the group, and establishes proper heading of the MAS, while not requiring communication or absolute position information such as GPS location. The follower group resembles a herd and cannot have any particular shape to it, but the offset amounts can be adjusted to tune the movement. This is the coordination method pictured previously in Figure 13.

5.4.2 Leader/Follower with Explicit Instructions

The second method for comparison was a leader-follower relationship with explicit instructions. This represents a more traditional, centralized control scheme: a single leader gives the other agents instructions for their next (relative) positions. Specific formations are prescribed to increase the efficiency and reach of the group. The agents are placed into a flock at the observation point to quickly move towards the target, then they are moved into a circle around the target when they are near it to maximize sensor coverage and prepare to follow erratic target movements. The gradient descent method was used for path planning at the agent level to guide both the leaders and followers to the intended next position, but the leader provided those waypoint positions to each of the follower agents. Figure 16 shows this method in the simulation.



Figure 16. Leader-Follower with Explicit Instructions.

5.4.3 Decentralized – Boids

Next, a fully-decentralized method based solely on boids was added for comparison. Upon reaching the target, agents will see it with their sensors and can pursue. Each agent simply aligns its heading with near neighbors while maintaining cohesion and separation based on specified ranges of acceptability. The group basically follows the target as a herd without requiring communication or any more than local sensing, as shown in Figure 17. This method gives the MAS an efficient passive movement capability, but it cannot assume complex formations or follow multiple targets.

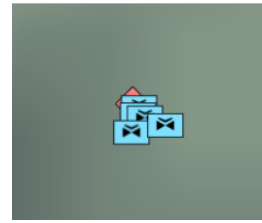


Figure 17. Decentralized – Boids MAS.

5.4.4 Leader/Follower with Subgroups & Boids

A fourth coordination scheme also started with a centralized, “leader-follower” style control and then separated the MAS into several subgroups to better handle multiple targets. When a leader sees new targets that it is not following, it first checks with any other swarms within communication range to see if they are following those targets. If not, the leader divides up its followers evenly (and randomly) into as many subgroup swarms as there are targets. Each new swarm is randomly assigned a leader, and they follow the same target tracking behavior described previously for leaders.

This method divides nearby MAS agents into relatively equal-size sub-swarms as new targets are seen, to ensure better sensor coverage as the target moves in unpredictable ways through the mission map. In this first implementation, the subgroups follow the leader simply using the boids method; however, explicit instructions could also be passed to the followers. There is still a single large MAS that includes all the agents with a single leader, which could allow for global “return to base” type commands—though this was not included in the current project’s programming. This method also blends two different control schemes, as a first step towards a more hierarchical and reactive control system.



Figure 18. Leader Subgroups with Boids.

5.4.5 Mission Stages and Assumptions

Some of these methods depend on conditional statements related to mission phases and conditions. As discussed in the mission description, the “stages” for this project are as follows:

- 1) At the observation point
- 2) Moving towards the target
- 3) Searching for the target
- 4) Tracking the target once it is found.

For control simulation purposes, these stages characterize the changing mission needs and guide the implementation of different formations and agent behaviors. At the observation point, the agents are considered at a friendly “base” location with good GPS signals and good communication. These conditions are not assumed for any of the other stages. However, during all stages, it is assumed that if the target or another agent is within a certain “near-field sensing” distance, it is positively “seen” and can be correctly identified as a friendly or foe entity. Furthermore, if a target is seen, it can be tracked. Initial target position is also assumed to be known, though some test cases do give the target initial movement from that known position. Future simulation experiments can add further uncertainty into this system, but this was outside the realm of this project.

6. URBAN MISSION TEST CASES

Table 2 below outlines the basic design of experiments (DoE) that was run using the UV-CoRE simulation environment with the urban mission previously discussed. A batch mode capability was used to allow for rapid, sequential test case execution. Agent numbers from one single agent up to ten were tested, as well as up to ten targets. Initial target behavior was also varied: Table 1 describes the three different initial target behaviors tested in this simulation. In all test cases, the targets were started at the Tech Tower GPS location, and the UAV agents were started at a specific North Avenue GPS location, as depicted in Figure 11.

Due to the simplicity of the path planning and obstacle avoidance routine used in this simulation, agents often got “stuck” when trying to track the target. This is considered a possible limitation of the simulation more than a result of the coordination methods. Therefore, additional tests were run with targets fleeing only outside and around buildings. In addition to this, stochastic tests were also conducted by perturbing a single target’s path by small amounts, to compare the coordination methods with fewer variables at once.

Table 2. UV-CoRE Urban Mission Test Cases.

Coordination Scheme	Agents	Target #	Target Initial Behavior
Leader-Follower & Boids	1, 2, 5, 10	1, 2, 5, 10	1, 2, 3
Leader-Follower & Explicit	1, 2, 5, 10	1, 2, 5, 10	1, 2, 3
Decentralized - Boids	1, 2, 5, 10	1, 2, 5, 10	1, 2, 3
Leader-Follower & Subgroups	1, 2, 5, 10	1, 2, 5, 10	1, 2, 3

7. URBAN MISSION TEST RESULTS

The various simulation runs show that the leader-follower method with boids is very effective at following a single target throughout the map. Whether the MAS starts out near to the target or relatively far, as long as the target does not change location before the MAS is nearby, there is 100% target tracking success. Each agent stays nearby, and all have good

eyes on the target. However, this is a rather wasteful use of many aircraft: the individual agents remain in the group and cannot follow a second target, even if one is seen. The follower agents also have no way to change their position relative to the target for more optimal views or better tracking. Furthermore, if the leader is compromised in any way, they would have no instructions. Even if logic is added such that the boids will follow the target(s) directly in this case, only one agent is really needed in most cases to adequately maintain continuous target tracking. This control scheme can be useful to quickly and simply move agents in Stage 2 of the mission, when there is a known target location relatively far away, but is not recommended for use where more complex movements or subgroups would be more effective.

The classic centralized control technique—the leader-follower with explicit instructions method—showed results that conformed to expectations but did not perform well overall in these test missions. Formations were possible both to get to the target as well as to track the target, and these would help to increase data gathering in the case of a fleeing person. However, in many test runs, some of the agents get left behind. They do not always have proper communication channels to find their way back to the leader, especially when a building is in the way and the rest of the MAS has moved far away. The follower agents heavily rely on the single leader for instructions, thus this method needs additional subroutines to handle situations where movement is constrained and communication is not guaranteed.

The boids decentralized technique was effective whenever the agents were within sensing distance of the target and for the duration of the tracking period. This method on its own performed similarly to the leader-follower with boids test case (Method 1), showing that having a leader with such a passive method is unnecessary. The boids method was much more successful than the centralized methods at bringing agents around complex obstacle shapes and continuing to follow the target. A robust control scheme could benefit from having this method implemented during MAS movement phases within a mission, as agents may be lost along the way, and boids would continue to move the remainder of the group along without trouble.

Finally, the subgroups “Method 4” mixed centralized and decentralized theory to form subgroups, if multiple targets were discovered during the mission. This method was the only one to successfully track multiple targets in any test case, albeit not very successfully in all cases. The simulations showed that agents are more likely to get left behind when they encounter complex obstacles; however, with more sophisticated path planning and obstacle avoidance routines, this should prove less of an issue. In general, this method requires more sophisticated AI schemes to correctly identify new targets, split the MAS into groups, assign new leaders, and then track a target while avoiding

obstacles. It is also vulnerable to the loss of subgroup leaders without any other backup methods implemented, but it is less vulnerable than any group with only one leader.

Comparing the methods in terms of the metrics proved challenging given the amount of agents that got “stuck” behind obstacles during the simulations runs. A sample of metrics output plots for each test case is shown in Table 3 on the following page. For all control methods, increasing the number of UAV agents has a clear benefit—especially with low numbers of targets. This adds strength to the argument that coordinated multi-agent systems are worth researching further. What is also readily apparent is that Method 4 does not perform as predictably well as the other methods as agents are added to the mission. This is likely due to the simplicity of the logic in this simulation, resulting in more agents getting stuck behind buildings. More complex path planning and obstacle avoidance AI could help groups of agents to more effectively move towards and track multiple targets. Overall, it can be said that the methods performed similarly from just looking at this data.

Additional stochastic test cases sought to clarify any differences between the methods for just a single target case. On Page 13, Figure 19 shows that the centralized leader-follower method with explicit instructions clearly performs the best in terms of single-target tracking, while the boids method is a bit less capable and the leader-follower scheme with just passive followers is the least effective. This result confirms the observations seen during simulation runs: the decentralized boids method is a very effective method for moving the group along after a target through a complex mission environment, and it is almost as good as the leader-follower method requiring constant communication. The smaller error bars indicate less variability in some results as the target’s path is perturbed slightly. The explicit instructions method sees the lowest variability (and thus the highest reliability) regardless of the number of agents, and this falls in line with expectations from research. However, the boids method alone also shows low variability, especially as the number of agents is increased: again, this indicates that the boids method may be very effective in some realistic scenarios. The mixed method with boids and leader-follower coordination performs poorly and unreliably overall, indicating that this combination is significantly less useful in robust control system design. In reality, this method likely just requires better AI and tuning to show its full potential.

8. CONCLUSIONS

The various multi-agent system control methods have shown distinct benefits and limitations as the mission complexity and size was varied. While the leader-follower techniques with explicit instructions may be susceptible to communication dropouts and sensor loss, the centralized methods form a reliable and effective basis for initial MAS coordination. With

adequate instructions pre-loaded into the leader, and with appropriate conditional statements to identify new stages of the mission, a leader-follower with explicit instructions MAS is the most effective target tracking technique.

Given the vulnerability of a leader-follower system in contested environments, other methods would be very useful as fallback coordination schemes. The boids method is very effective when the agents are relatively close together, such as at the start of the mission. This method could also be used to continue tracking a nearby target right after a leader is eliminated, to ensure that the chain of custody is not broken despite complex maps and other interfering factors. However, the boids method is not the most efficient use of many agents, as they all simply follow in an uncoordinated group.

For the cases where multiple and complicated targets exist, a leader-follower scheme with explicit instructions and subgroups formed as the targets are discovered has the potential to be very effective in continuously tracking of each of the targets—up to the number of agents available (minus one). More complex instructions and logic are necessary to optimize this method further, as the simple version used in this simulation did not perform well when the metrics were calculated. Depending on mission needs, these results can guide the creation of a hierarchical control architecture comprised of a blend of the aforementioned methods, which will be as reactive and robust as needed for real mission scenarios.

Finally, results were generally in line with expectations for each method, showing that the UV-CoRE simulation environment is capable of modeling MAS coordination schemes for a complex urban environment.

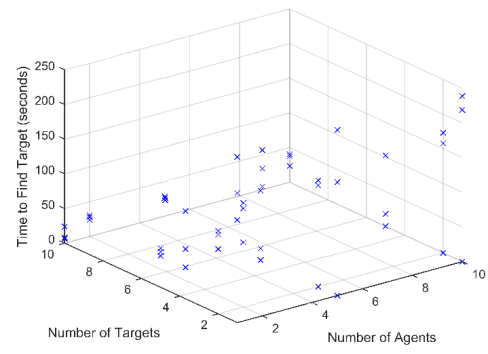
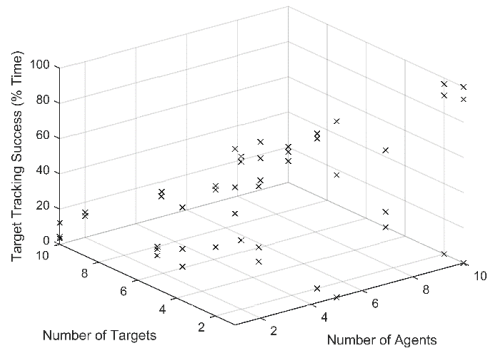
9. ACKNOWLEDGMENTS

The authors would like to thank Dr. Kelly Griending and Dr. Dimitri Mavris for their guidance throughout the development of this project, as well as, Mr. Eric Zellers, Mr. Kyle Motter, Dr. Tsiotras, Dr. Eric Johnson, and Mr. Brett Amidon for their valuable background information and review of this work.

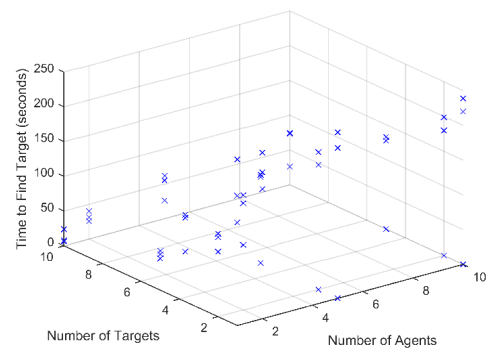
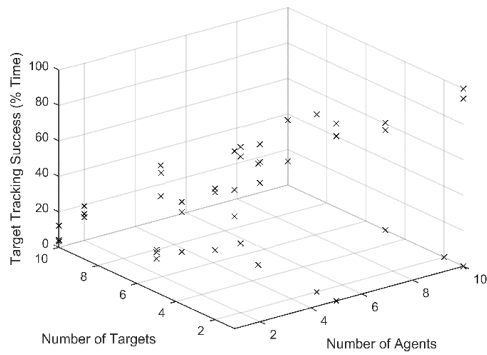
Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ERF proceedings or as individual offprints from the proceedings and for inclusion in a freely accessible web-based repository.

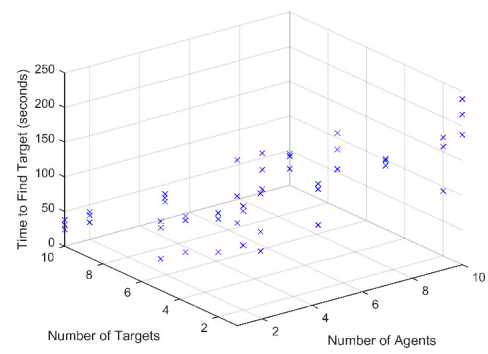
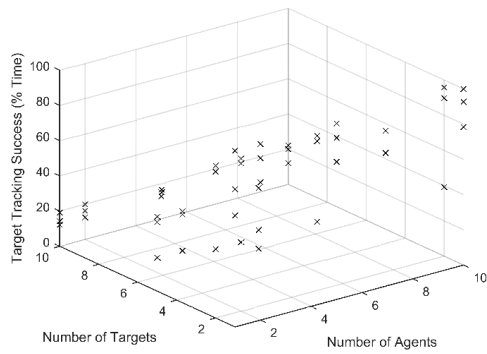
Table 3. UV-CoRE Urban Mission Simulation Results from Full DoE.



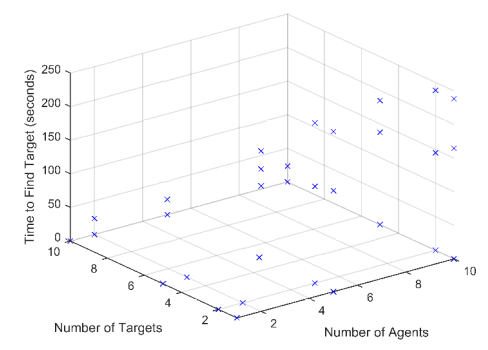
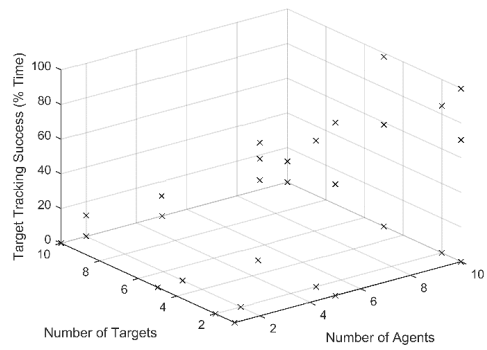
Method 1: Leader-Follower with Boids



Method 2: Leader-Follower with Explicit Instructions



Method 3: Decentralized - Boids



Method 4: Leader-Follower with Subgroups Using Boids

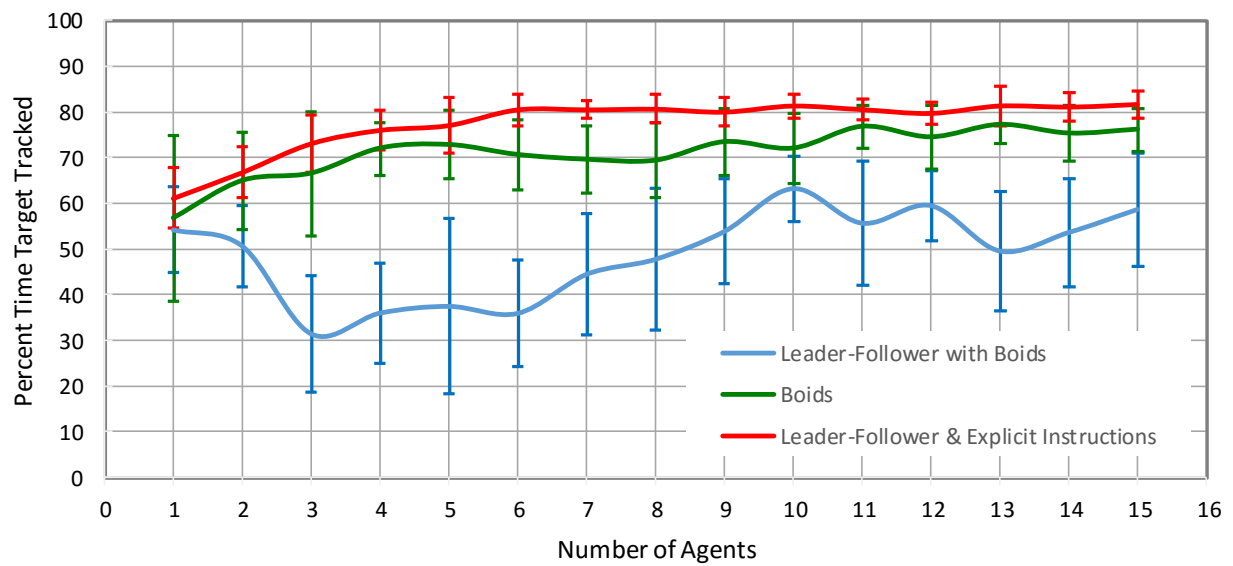


Figure 19. Results of Single-Target Stochastic Testing.

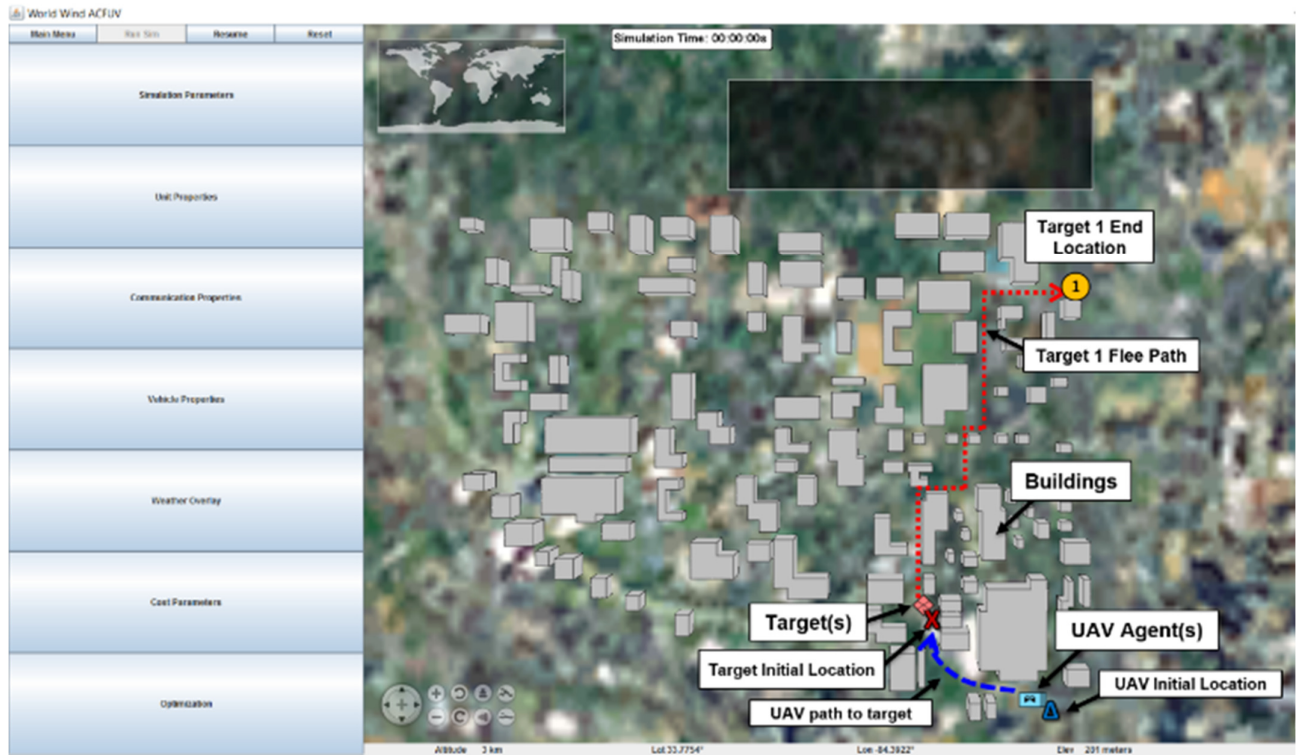


Figure 20. UV-CoRE Simulation Environment: Larger View of Simulation GUI.

REFERENCES

- [1] "More than just information gathering: Giving commanders the edge," May 26, 2014, NATO, www.nato.int
- [2] Aeryon Labs, Inc. www.aeryon.com
- [3] "How Libyan Rebels Got A \$120,000 Micro-Drone – Forbes," www.forbes.com, Aug 26, 2011.
- [4] Egerstedt, M., "Control Theory and Swarm Robotics", Lecture, Intro to Robotics AE7785, Atlanta, GA, August 21, 2015.
- [5] U.S. Coast Guard Addendum to the US National Search and Rescue Supplement (NSS) to The International Aeronautical and Maritime Search and Rescue Manual (IAMSAR) (COMDTINST M161). January 7, 2013.
- [6] K. Stewart, "ARSENAL Breaks Previous Records, Flies 20 Autonomous UAVs," June 11, 2015, NPS, www.nps.edu
- [7] Scharre, P, Robotics on the Battlefield Part II: The Coming Swarm, 2014.
- [8] D. Smalley, "LOCUST: Autonomous Swarming UAV's Fly Into the Future," April 2015, ONR, www.onr.navy.mil
- [9] L. Elkins, D. Sellers, and W.R. Monach, "The Autonomous Maritime Navigation (AMN) project", Journal of Field Robotics, vol 27, 2010, pp. 790–818.
- [10] M. Rubenstein, A. Cornejo, R. Nagpal, "Programmable Self-Assembly in a Thousand-Robot Swarm", Science, Vol 345, no 6198, 15 Aug 2014.
- [11] Tsiotras, P. et. al., "Optimal Trajectory Planning and Assignment Problems for Kinematic Agents with State-Dependent Metrics", Presentation at MIT.
- [12] Christmann, H. C., A Path Planning Aid for Single-Operator UAV Swarms in Structured Environments, Doctoral Thesis, Georgia Institute of Technology, August, 2015.
- [13] Christmann, H. C., Self-Configuring Ad-hoc Networks for Unmanned Aerial Systems, Master's Thesis, Georgia Institute of Technology, May, 2008.
- [14] Wang, Y., Chen, P., and Jin, Y., Trajectory Planning for An Unmanned Ground Vehicle Group Using Augmented Particle Swarming Optimization in a Dynamic Environment, IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, October, 2009.
- [15] Wang, Qiang, et. al., Flocking control for multi-agent systems with stream-based obstacle avoidance, Transactions of the Institute of Measurement & Control, 2013.
- [16] Aksaray, D., "Autonomous Hopping Rotochute", MS Thesis, March 2011, Georgia Institute of Technology, Atlanta, US.
- [17] D. Aksaray and D. Mavris, "Maintaining Connectivity for Networked Mobile Systems in the Presence of Agent Loss", AIAA Guidance, Navigation, and Control Conference, Boston, MA, US, Aug. 2013.
- [18] D. Aksaray, Formulation of Control Strategies for Requirement Definition of Multi-agent Surveillance Systems, Dissertation, Georgia Institute of Technology, 2014.
- [19] Boyle, Ashely, "The US and Its UAVs: A Cost-Benefit Analysis." American Security Project. 24 July 2012.
- [20] Harvard University, "A self-organizing thousand-robot swarm." Aug 14, 2014, <http://wyss.harvard.edu/>.
- [21] M. Ji, "Distributed Coordination Control of Multi-agent Systems While Preserving Connectedness," IEEE Transactions on Robotics, VOL. 23, NO. 4, AUGUST 2007.
- [22] Department of Defense Interface Standard: Common Warfighting Symbolology (MIL-STD-2525C). 17 November 2008.