10th International Command and Control Research and Technology Symposium The Future of C2

Moving Emergent Behavior Algorithms from Simulation to Hardware: Command and Control of Autonomous UxV's

> Chad Hawthorne Dave Scheidt

The Johns Hopkins Applied Physics Laboratory 11100 Johns Hopkins Road Laruel, MD 20723-6099 Phone: 443-778-5000 Fax: 443-778-6667 chad.hawthorne@jhuapl.edu

Abstract

The future unmanned battlespace will contain heterogeneous swarms of autonomous air and ground platforms, with individual platforms coming in many different flavors, from stationary seismic sensors, to mobile acoustic sensors, to airborne visual sensors. A significant hurdle in enabling a heterogeneous swarm is the ability to move the algorithms developed in simulation environments onto real-world unmanned vehicles (UxV's). To enable this capability, The Johns Hopkins University Applied Physics Laboratory (JHU/APL) has developed the Robotic Algorithm and Communications Environment (RACE), a platform independent behavior-based algorithm framework that supports air and ground vehicle hardware interfaces to enable swarms of UAV's, UGV's and simulated vehicles to operate cooperatively.

Hardware in the loop experiments have been carried out with mixed ground and air vehicles. These tests were used to validate the emergent behavior algorithms that were developed in simulation. A full spectrum of behaviors was tested on both air and ground vehicles including: open area searching, searching of road networks, searching of densely cluttered areas, recruitment for the purpose of classification, perimeter protection and pursuit. JHU/APL provided C^2 capabilities for the swarm of vehicles through two separate user interfaces, requiring the swarm to de-conflict disparate goals, and self-organize along tasking lines. Specific elements used in the tests included four ground vehicles, two air vehicles, two unmanned ground sensors, two operator workstations, six "buildings" and three distinguishable non-swarming mobile objects.

The demonstration was developed using RACE, which runs on a variety of platforms including various flavors of PC's, UxVs, and handheld devices. Additionally, it provides a vehicle abstraction layer, allowing the AI to execute independent of the specific robotic platform. This paper will describe the C^2 aspects of the RACE architecture and our results from recent hardware in the loop demonstrations. Our experiments have shown how swarms of autonomous vehicles can support complex C^2 environments by cooperatively de-conflicting multiple user goals.

Introduction

The overwhelming majority of military commanders throughout history, including Alexander, Napoleon, Lee, Pershing, Patton, Schwarzkopf and Franks, have used hierarchical Command and Control (C^2). Hierarchical C^2 aggregates information as it moves up the command chain to provide front-line soldiers with detailed local knowledge and top level commanders with abstract global knowledge. Likewise, courses of action are constructed globally at the highest command level and are decomposed at each subordinate layer of command. Interactions between peer units within the organization are tightly constrained by the superior command level. Mission success depends upon reliable communications between successive layers in the command hierarchy.

It is possible for military organizations to achieve organization-wide effectiveness without the use of hierarchical C^2 . The alternative to hierarchical C^2 is heterarchical C^2 , in which control is decided through local peer-to-peer interactions. While less common than hierarchical C^2 the heterarchical approach has been used successfully throughout history. A strength of heterarchical C^2 is enhanced pace of operations, robustness and survivability provided by opportunistic collaborations between ad hoc members of the force; that is, combatant cooperation occurs unexpectedly whenever an opportunity presents itself. One historical example of these types of military maneuvers is the use of swarming tactics [1]. Swarming tactics, in which a mass attack is made on an enemy position, have been used successfully by Ghengis Khan, Napoleon during the Ulm Campaign, the Japanese in their Kamikaze Attacks, the Germans in the Battle of the Atlantic and by the Somalis in Mogadishu. Additionally, research has shown that organizations operate most efficiently when their structures and processes match their mission environments [2]. As these examples illustrate, the most effective swarming tactics were executed within a heterarchical C^2 environment that matched the swarming organizational structure.

In the last decade substantial theoretical progress has been made in modeling the emergent behavior found in heterarchical C^2 organizations. Two significant theoretical avenues have been pursued: Swarm Behavior, which derives its motivation from communal species found in nature, and Cellular Automata, which traces its roots to computational theory. Both approaches use simple local interactions to generate complex emergent group behavior; behavior that is used to satisfy group objectives without requiring global knowledge or command. Recently, military visionaries have begun to investigate military uses of unmanned vehicle "swarms" using emergent control [3] [4]. This is because, from a military perspective, swarm behavior exhibits several appealing traits:

- Swarm behavior has been shown to provide high quality results to intractable problems.[5]
- A swarming group's ability to perform a task is independent of the size of the organization.
- The decision loop, thereby the intelligence-to-trigger time, is less than that in hierarchical C² systems.
- The group as a whole is more survivable than hierarchically controlled groups.

This paper describes the C^2 environment developed by JHU/APL for hardware-in-theloop experiments in heterarchical UxV control. This paper details the architecture for controlling and simulating a swarm of vehicles in a heterarchical C^2 environment. The swarming algorithms used in these experiments are described elsewhere. [6]

Unmanned Vehicle Automation

The future unmanned battlespace will contain heterogeneous swarms of autonomous air, ground, waterborne, submarine and subterranean platforms, with diverse sensor payloads including electro-magnetic, electro-optical, acoustic and seismic sensors. The

proliferation of these unmanned vehicles will produce an enormous amount of information for users and systems to process. In the DoD's vision of network centric warfare [7], the network will provide connectivity between all of these nodes, allowing each sensor and UxV to serve as a client of other systems on the network. As the connectivity of the network increases, so does the amount of information potentially available to members of the network. Without a significant increase in automation, network members will quickly become overloaded with information. Automation comes in many different levels. The standard description of these automation levels is the scale defined by Sheridan and Verplank [8].

Figure 1 Sheridan and Verplank's Levels of Automation

Automation can be divided into three broad categories, human-in-the-loop, human-onthe-loop, and human-out-of-the-loop. Today's tele-operated unmanned systems are primarily human-in-the-loop, requiring an operator to approve or direct vehicle actions. These vehicles typically operate within the existing military hierarchical C^2 environment, with an operator on the ground serving as a proxy for the autonomous vehicle. Even vehicles such as Globalhawk that do "hand off" vehicle operations between operational units do so in accordance with pre-arranged agreements made within the hierarchical command structure. Truly autonomous vehicles that operate heterarchically, taking highlevel direction from operators and cooperating with other vehicles that are not defined at launch do not yet exist.

Sheridan and Verplank's automation levels are useful for describing these control paradigms; automation levels one through five describe human-in-the-loop control, in which a dedicated operator is required for an automated system to function. In these levels of automation, tasks can only be accomplicate to the paradigment of the execute his/her decision cycle.

Level

Human-on-the-loop control can be described as automation levels six through nine. In these levels of automation, operators are removed from the control loop, allowing autonomous systems to function with infrequent or no human intervention. Swarming software developed by JHU/APL supports this human-on-the-loop level of automation, in which the user provides high-level goals to the swarm. By removing the human from the control loop, we can support a faster decision cycle, reduce manning requirements, and support a heterarchical command and control environment.

Swarm Situational Awareness

For swarming algorithms to operate in this heterarchial C^2 environment, each entity in the swarm must maintain its own local view of the world and act upon it without explicit direction from a hierarchical command structure. A high-level model that maintains the global situational awareness does not exist in a heterarchical organization, so each vehicle must maintain its own local model. Situational awareness is defined as having three levels, 1) Perception of the Environment, 2) Comprehension of the Situation, 3) Projection of Future States, or Reaction to Stimuli [9]. The diagrams below show these three levels as well as how our unmanned vehicles achieve these levels of situational awareness.



Figure 2 Levels of Situational Awareness (adapted from Endsley)

Traditionally, situational awareness meant that the acting entity had a nearly complete picture of the environment in its sphere of influence or area of activity. This paradigm assumed that the environment could be sensed and objects of interest extracted in a timely manner. Assuming successful identification and classification, the machine would create a virtual world that represents the sensed real world. The problems with this approach is that for most environments sensing is difficult, classification is time consuming and information about the environment is incomplete [10].

To solve the problem of modeling an uncertain and dynamic environment, we have followed the philosophy promoted by Brooks [11], "The world is its own best model". In this approach, a completely accurate global model of the world is not maintained; instead

a simplified local model sufficient to complete the automated control loop is built. This shift allows simplification of sensing and identification processes and eliminates the need to maintain a global representation. By simplifying the identification process and maintaining local situational awareness, each vehicle can operate completely autonomously, allowing the swarm to function without hierarchical control while achieving mission goals through self-organization among peers.

Command and Control

JHU/APL's autonomy software supports human supervisory control and fully automatic control of the swarm of vehicles. Human supervisory control, as defined by Sheridan [12], means that an operator is responsible for providing intermittent programming objectives to a computer. The computer closes an autonomous control loop to carry out the system objectives. In our system, the computer is not a single machine, but a swarm of vehicles for which the user monitors the effectiveness and explicitly provides high-level goals.

In addition to purposefully expressing operational goals, human operators can implicitly influence the behavior of the swarm by his/her own actions, providing fully automatic control of the swarm. For example, we conducted a hardware experiment where a convoy entered an area patrolled by a group of UAV's. The swarm of UAV's detected the presence of the convoy and provided video surveillance to the convoy operator. This occurred without any explicit input from the operator; the operator interacted with the swarm by presence and was not required to supervise or monitor the swarm of vehicles.

The following diagram [12] shows the three types of operator control, from fully manual control, to supervisory control, to fully automated control. As described previously, our system supports both supervisory control and fully automatic control.



Figure 3 Levels of Operator Control

In each of the control modes (supervisory and automatic), JHU/APL's autonomous vehicle swarm supports multiple human operators. Each operator can provide high level goals to the swarm, either through their mere presence or more explicitly through an operator interface. The swarm is responsible for de-conflicting user goals and self-organizing along tasking lines to accomplish the mission. Communications in the swarm happens opportunistically, meaning that the goals provided by an operator spread out through the swarm as communication allows; this heterarchial organization allows the swarm to function with lower overall bandwidth and operate in dynamic, uncertain environments.

Software Architecture

When developing emergent behaviors, simulation of the behaviors in numerous operational environments is required to effectively characterize the behavior of the swarm. We developed our Robotics Algorithm and Communications Environment (RACE), to run on both real-world vehicles and in a simulation environment, allowing quick turnaround from the lab to the field. Additionally, RACE allows simulated vehicles and real-world hardware to operate concurrently in a swarm.

To support these requirements, we followed three primary design principles. 1) the swarm must be decentralized, 2) the software should be independent of the robotic

platform, and 3) the software should be independent of sensor types. By making the control software independent of hardware allowed us to run simulated vehicles using the identical software as were run on the UxV's.

There are three logical modules in RACE. The first module, the Beliefs module, is responsible for maintaining the local situational awareness for a vehicle. The Belief module must manage and de-conflict information about the external world received from on-board observations and from peers. This module is also responsible for receiving and de-conflicting high-level user goals. The second module, the Behavior module, is responsible for executing behaviors to operate the vehicle based on tasking goals. Finally, RACE supports multiple platforms by abstracting the platform specific hardware control into a third module, the Actions module. This allows the same AI software to execute either in simulation, on UGV's, or on UAV's. The following diagram shows how these three components interoperate.



Figure 4 RACE software components

JHU/APL Tests

Hardware in the loop experiments have been carried out with mixed UGV's and UAV's using the RACE architecture. A full spectrum of behaviors was tested on both air and ground vehicles including: open area searching, searching of road networks, searching of

densely cluttered areas, recruitment for the purpose of classification, perimeter protection and pursuit. JHU/APL provided C^2 capabilities for the swarm of vehicles through two separate user interfaces, requiring the swarm to de-conflict disparate goals, and selforganize along tasking lines. Specific elements used in the tests included four ground vehicles, two air vehicles, two unmanned ground sensors, two operator workstations, six "buildings" and three distinguishable non-swarming mobile objects. Figure 5 shows one of the UGV's used in the multi-vehicle control experiment.



Figure 5 UGV used in Multi-Vehicle Control Experiment

All four robots used short range fixed acoustic sensors and SICK laser range finders for obstacle detection and avoidance. Localization for both ground vehicles was accomplished with commercial GPS receivers. The iRobot Mini used a fixed long-range directional microphone to classify non-swarming objects. Robot control was processed using on-board Pentium-class microprocessors running the Linux operating system underneath iRobot's Mobility software.

The unmanned air vehicles were modified remote control air vehicles with approximately six-foot wingspans. As with the ground vehicles, air vehicle localization was accomplished with GPS receiver. A commercial autopilot by MicroPilot was used for flight control. For safety reasons the RACE software was not run on-board the aircraft. Rather autonomy ran on-board a MicroPilot groundstation. RACE software received GPS telemetry from the MicroPilot groundstation and, based upon that telemetry, provided the MicroPilot Ground Station with a desired waypoint that was a fixed distance away from the air vehicle in the direction of the movement vector. The ability to perceive objects in the real world was a key factor in demonstrating the swarm behaviors. Because the focus of this research was on control rather than perception the unmanned vehicles detected moving objects with virtual sensors. Virtual sensing was implemented via a GPS enabled wireless personal assistant or laptops on board each moving object. Each object would then announce its position to those vehicles that were within communications range.

Tests were conducted in accordance with the following plan. First two separate areas were defined in which a target of interest might be located. One area consisted of the

immediate vicinity of the buildings; a second larger area consisted of a larger rectangular area. Simultaneously a third objective was defined; the protection of a building central to the town. Separate operators provided each of these objectives. The swarm of vehicles was responsible for self-organizing along tasking lines to accomplish all user goals.

The ground vehicles responded to these objectives by first searching through the town for objects. After finding no targets the ground vehicles then formed a slowly revolving circular perimeter around the protected building. In parallel, the air vehicles performed a continuous patrol of the large rectangular area. In time, the mini-van object entered the test area. Upon detecting the mini-van's presence the air vehicles broke off their patrol and established a revolving circular air cover over the mini-van. Circular air cover was maintained over the mini-van throughout a number of tests and maneuvers. The two human objects were stationed at a crossroads along the minivan's route, representing a "crowd" of two. As the mini-van approached the intersection the air vehicles detected the existence of both human objects. The acoustically enabled iRobot Mini, upon learning from the air vehicles of the existence of unclassified objects, broke formation and pursued first the closest, and then the farthest of the human objects, classifying them in turn. Upon receiving data announcing the existence of a classified human object of interest.

These behaviors were carried out fully autonomously by the swarm of vehicles on multiple occasions. The only input required by the operator was the high level search areas and assets to be protected. Once the goals were entered by the user, the swarm of vehicles was able to self-organize and accomplish a complex mission with minimal operator intervention.

Conclusion

As vehicles with greater autonomy are fielded operationally, the C^2 environment that the vehicles operate in must be explored. Increased autonomy of unmanned vehicles will eliminate the need for direct operator control over vehicles; however, operator input and human interaction must be well-understood before fielding swarms of autonomous systems. In the case of swarming behaviors, we believe the most effective C^2 environment is heterarchical organization. Although more experiments are required, initial results have shown that swarms of autonomous vehicles can support a heterarchical environment through local communication, cooperatively de-conflicting user goals, and self-organization.

References

- 1. *Swarming and the Future of Conflict,* John Arquilla, David Ronfeldt, RAND report DB-311-OSD, 2000.
- Levchuk, G.M., Lebchuk, Y.N., Luo, J., Pattipati, K.R., and Kleinman, D.L., "Normative Design of Organizations – Part I Mission Planning." IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 32(3), 346-359., 2002.
- 3. Becker, J. A Buzz in the Air, Intelligence Surveillance & Reconnaissance Journal March, 2003, pg. 22-24.
- 4. Software for Distributed Robots, http://www.darpa.mil/ipto/research/sdr/index.html
- 5. Eric Bonabeau, Marco Dorigo and Guy Theraulaz *Swarm Intelligence: From Natural to Artificial Systems* New York, NY: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity 1999
- D. Scheidt, T. Neighoff, R. Bamberger and R. Chalmers, Cooperating Unmanned Vehicles, AIAA Intelligent Systems Technical Conference. Chicago, IL, Sep. 21, 2004.
- 7. DoD, C.I.O, "Network Centric Warfare Department of Defense Report to Congress." Office of Secretary of Defense, Washington, D.C., 2001.
- 8. Sheridan, T.B., and Verplank, W.L., "Human and Computer Control of Undersea Teleoperators", MIT, Cambridge, 1978.
- 9. Endsley, M. "Design and Evaluation for situation awareness enhancement. Paper presented at the Human Factors Society 32nd Annual Meeting, 1988.

10. Ko J., Mahajan A., Sengupta R. A Network-Centric UAV Organization for Search and Pursuit Operations. Proc. of the 2002 IEEE Aerospace Conference, 2002.

- 11. R. Brooks "The Whole Iguana" In Robotics Science, ed. M. Brady, MIT Press, Cambridge, MA, pp. 432-56. (1989).
- 12. Sheridan, T.B. "Telerobotics, Automation and Human Supervisory Control", The MIT Press, Cambridge, MA.