

# Flight Demonstrations of Unmanned Aerial Vehicle Swarming Concepts

*Robert J. Bamberger Jr., David P. Watson, David H. Scheidt, and Kevin L. Moore*

**A**PL has been engaged in a number of independent research and development projects over the past 5 years intended to demonstrate the cooperative behaviors of swarms of small, autonomous unmanned aerial vehicles (UAVs). Swarm members cooperate to accomplish complex mission goals with no human in the loop. These projects represent a variety of approaches to UAV swarming, including teaming, consensus variables, and stigmergic potential fields. A series of experiments was conducted from 2001 through 2005 to demonstrate these concepts, and research in this area is ongoing. As a result of these efforts, APL has developed autonomy frameworks, hardware architectures, and communications concepts that are applicable across a broad range of small, autonomous aerial vehicles.

## INTRODUCTION

Effective employment of autonomous unmanned assets meets a critical need in today's military. These assets can perform crucial missions in denied or dangerous areas while keeping the warfighter out of harm's way. Until recently, the unmanned aerial vehicles (UAVs) used for these missions typically have been battalion-level Class III or company-level Class II UAVs, which are costly and too large for small units. Furthermore, scheduling of these resources typically requires coordination far up the command structure.

More recent military operations in Iraq and Afghanistan have demonstrated the utility of organic assets that can be deployed by "boots on the ground" at the battlefield edge. These small, platoon-level Class I UAVs are inexpensive (and in some cases expendable), are often back-packable, and typically require little setup time or logistical support. Small UAVs can be used to track

vehicles, enable communications, capture signals, exfiltrate sensor data, and in their most common role, provide the warfighter a bird's-eye view of the battlefield. Individually, their effectiveness is somewhat limited, but cooperating as a swarm, these assets can act as a distributed sensor system, employing sensors of various types and resolutions and providing different views simultaneously.

Most UAV systems, however, whether large or small, are manpower-intensive, relying on relatively high duty cycle remote control by operators. Operating under this paradigm, swarms of remotely controlled UAVs require as many skilled pilots as there are swarm members, and these pilots must be able to deconflict airspace demands, mission requirements, and situational changes in near real time.

The autonomy concepts being investigated by APL take the human out of the loop in terms of individual

vehicle control. With these systems, the operator provides high-level goals, constraints, and resources, relying on the system itself to deconflict, elaborate, and choose among alternative courses of action in mission execution. Simple launch and recovery, as well as low maintenance, are also major advantages to these assets. Trained pilots are not necessary. Notionally, for vehicles that can autonomously take off and land, a single soldier or first responder could service an entire swarm, primarily in a tasking, retasking, and refueling capacity. The uniformed services are also studying concepts wherein a number of small vehicles are deployed from larger UAVs, manned aircraft, submarines, ships, or guns.

APL has been investigating several approaches to the coordinated behavior of UAV swarm members. These concepts include a simple teaming arrangement, the use of consensus variables, and the use of stigmergic (i.e., biologically inspired) potential fields. A key component of these investigations has been ongoing flight tests, which are being used to develop implementation strategies for migration to operational systems, as well as demonstrate these concepts in actual flight.

## SIMPLE TEAMING APPROACH

A simple teaming approach was investigated in APL's initial UAV cooperative autonomy effort. The general goal was to have the means to accomplish complex missions through the cooperation of small, inexpensive vehicles of limited capability. Hence, the control system also had to be inexpensive and simple enough to require only a relatively small amount of computational power. At the same time, it had to be flexible enough to adapt to a dynamic environment. The integrated payload supported a "plug-and-play" sensor architecture and compatible subsystem interfaces while satisfying weight, power, and volume constraints.

A UAV independent research and development effort was initiated in 2001 to demonstrate this team approach. There were three primary technical objectives.

1. *Demonstration of multivehicle cooperative autonomy.* Specifically, flight tests were conducted to demonstrate the execution of a specified reference mission using two small, cooperating UAVs.
2. *Development of a simple, easy-to-use vehicle architecture.* A small, robust airframe was selected; plug-and-play payload modules were integrated; and simple launch and recovery procedures were developed.
3. *Demonstration of system integration.* The total system integration effort combined the flight segment (airframe, autopilot, mission control processor, sensors, and communications) with a ground segment that provided mission planning and operations tools, a user interface, and ground communications.

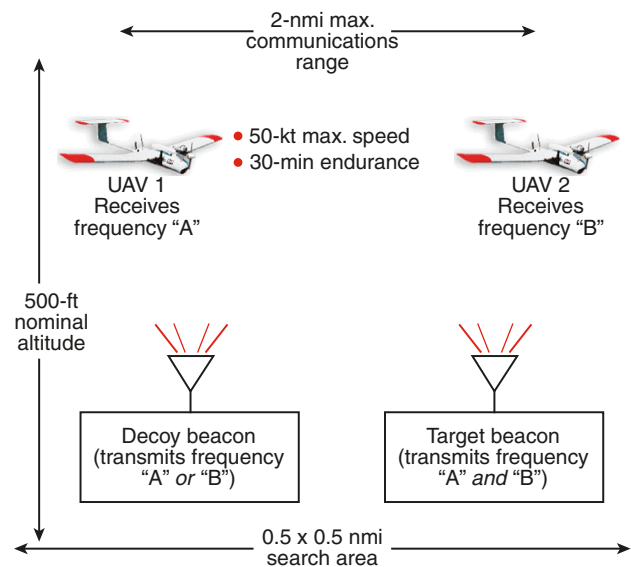
To demonstrate the teaming approach, a simple reference mission was developed: a team of two autonomous

UAVs tasked to cooperate to search for, locate, and positively identify a target, in this case an RF beacon. Each UAV was equipped with an RF detector tuned to a different frequency. Decoy beacons transmitted only a single tone, but the target beacon transmitted both tones (Fig. 1). Hence, to discriminate the target beacon from the multiple decoy beacons required cooperation of both vehicles.

Before launch, a human operator assigned both vehicles a basic mission (i.e., find the target) with the following tasks:

1. Following launch and flyout, the vehicles begin a search of a prescribed area according to an initial script.
2. When one vehicle detects a tone, it localizes and records the location of the possible target.
3. The identifying UAV then requests the second UAV to break from its search pattern and fly to that location to confirm or refute the identification as the target.
4. If the second UAV indicates that the beacon is a decoy (i.e., the second UAV does not detect a tone), then both UAVs return to their search patterns.
5. If the second UAV indicates that the beacon is the target (i.e., both UAVs detect a tone), the UAV equipped with a video camera loiters and captures images of the target, then both vehicles return home (the launch area) to be recovered.

While the search pattern was preprogrammed, target locations were not. Success required cooperation between the two vehicles, which had to be prepared to deal with certain unpredictable factors such as target location and detection sequence. Also, once the UAVs received the mission definition, the mission itself was



**Figure 1.** UAV simple teaming approach flight demonstration reference mission.

carried out without a human providing instructions or flight control.

### Mission Control Software

The challenge of creating the mission control software was that it had to be simple enough to require only the relatively small amount of computational power (e.g., no more than that of a personal digital assistant) that would be available on a small computing platform, while at the same time it had to be sufficiently flexible to adapt to a dynamic environment. The mission control software was responsible for five primary functions: (1) receiving mission definitions from ground commanders, (2) receiving processed information from the sensors (i.e., received frequency information from the RF sensors, geospatial location information from the flight controller, and platform condition data from onboard monitors), (3) exchanging messages with other vehicles and the ground station, (4) determining the necessary changes in the state of the vehicle, and (5) sending appropriate commands to the sensors and autopilot.<sup>1</sup>

Collaborative, autonomous behaviors were achieved by implementing a control architecture based on a simple framework. Many of the complex, time-evolving mission requirements could be constructed as a sequence of basic tasks and changing states. These basic tasks included search, loiter, send message, process sensor data, etc. Transitions between states could occur in response to

receiving communications from another team member, reaching a predetermined spatial location, detecting a vehicle health problem, sensing some phenomenon in the environment, or receiving a change of mission from the ground commander.

Mission specifications in the UAV control framework were represented as finite state automata (FSA). As shown in Fig. 2, FSA specifications comprise three types of components—states, transitions, and actions—that provide all the information required to execute a planned mission and respond to contingency events if necessary. FSA frameworks provide an intuitive and well-understood approach to implementing complex discrete control systems. In their most restrictive form, all actions are determined by the currently active state at any point in time. That state represents the history of the system in terms of its input since initialization. State transitions are expressed in terms of events that might be generated by mission situations (e.g., arrival at a waypoint), environmental conditions (e.g., excessive headwind), or vehicle parameters (e.g., fuel level below threshold). FSA systems provide a formal model that is amenable to verification and optimization and as a result have been widely adopted for the design of embedded systems, digital logic devices, and compilers. Even more compelling from a system implementation perspective is that FSA models can be directly mapped into executable code for the mission controller. In fact, it is relatively straightforward to create an implementation that executes the models directly at run-time, thus eliminating the requirement for manually coding mission software from a behavior specification, a potential source of subtle system implementation errors. Although

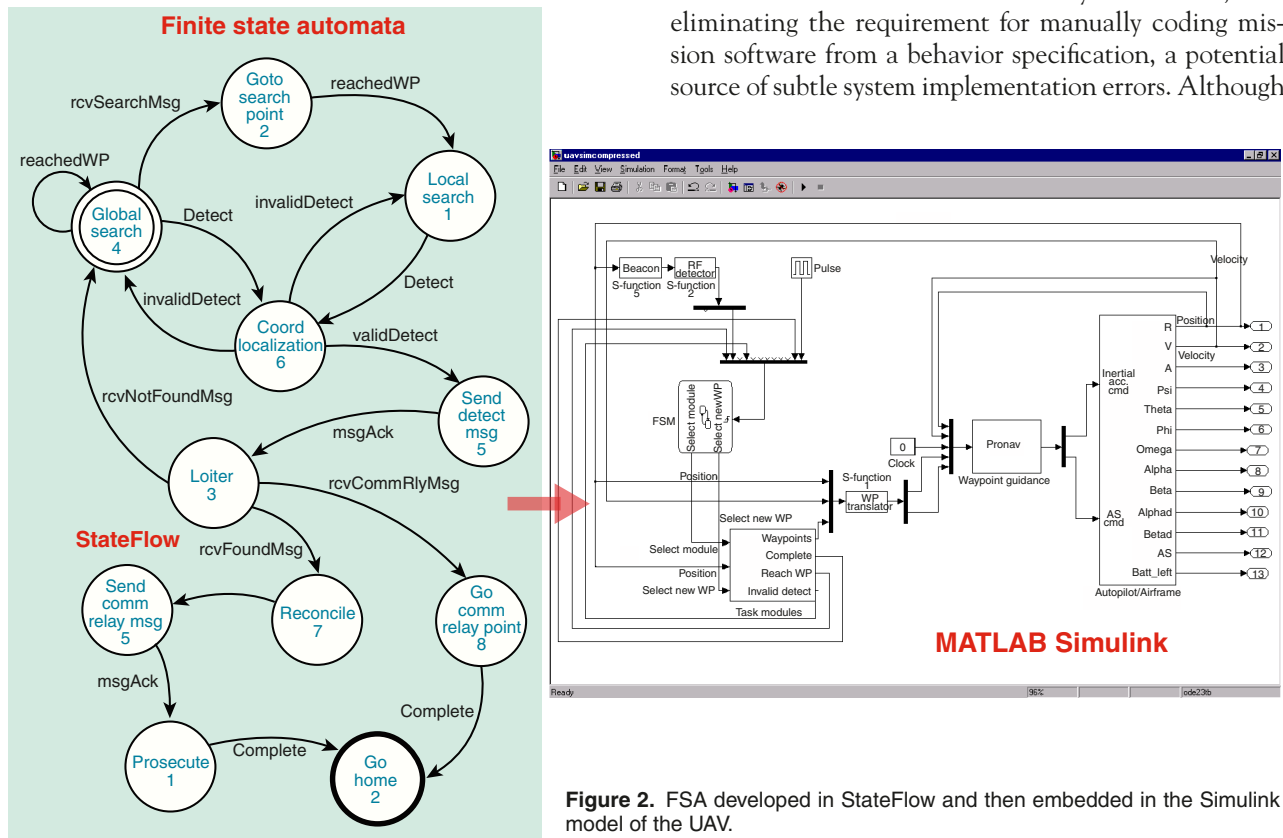


Figure 2. FSA developed in StateFlow and then embedded in the Simulink model of the UAV.

current behavior specifications for the UAVs are relatively simple, our objective was to create a mission control framework that could scale to higher levels of complexity where issues such as formal verification become more important.

## Hardware

Vehicle system requirements for this mission included a low-level controller for autonomous navigation and vehicle flight control; a wireless communication device for exchanging information between the two vehicles, as well as between the vehicles and ground station; a high-level controller used to interface with ground commanders, manage sensor data, enabled multivehicle cooperation, and execute the FSA; location sensors to determine vehicle spatial coordinates; and environment sensors to sense external phenomena (in the case of this mission, RF).

The vehicle used for the flight demonstrations was a fixed-wing Javelin UAV provided by BAI Aerosystems (Fig. 3). For low-level vehicle control, a GPS-enabled BAI Aerosystems proprietary autopilot was used. Vehicle-to-vehicle wireless communications were accomplished using 115.2-kbaud wireless modems. Vehicle communications, which included scheduled position updates, were monitored by the ground station. The ground station was used to load the mission on the vehicles, download diagnostics, and collect test metrics, but not to control the vehicles.

For this particular reference mission demonstration, each of the two UAVs had an RF sensor tuned to a different frequency. In addition to these RF sensors, one UAV was equipped with a small, low-resolution video camera to collect images of the target once it was identified. To support the plug-and-play system architecture, each sensor was coupled with a sensor interface module, which mapped sensor raw data to a serial bus protocol developed by APL. This configuration allows integration of new and varied sensors with no, or only minor, modifications to the mission control software.

The mission control software was implemented by the high-level controller, which also supported the plug-and-play sensor architecture, subsystem interface bus,

and communications module. These high-level control functions were executed on a single-board computer developed by APL to satisfy the vehicle weight, power, and volume constraints.

## Reference Mission Demonstration Results

Testing of the teaming concept consisted of software simulation, hardware-in-the-loop (HIL) tests, and ultimately flight tests. Simulations of the reference mission were first conducted using software representations of aircraft dynamics, autopilot operation, sensor outputs, and communications. These tests were accomplished to validate the FSA and their implementation.

After successful completion of the software simulation, HIL tests of increasing complexity were conducted using flight hardware and software, including the ground station. The flight tests also increased in complexity, starting with tests to demonstrate the ability of the vehicles to accept the mission definition, navigate using GPS waypoints, and fly the mission search patterns.

Although the final demonstration of the reference mission called for two UAVs in the air, because of the unavailability of equipment, only one airborne UAV was used. With one airborne vehicle and one vehicle emulated on the ground, the teaming behaviors were demonstrated, and the reference mission was successfully accomplished. This included the correct identification of decoy beacons and cooperation of the two UAVs to identify the target beacon.

## APL UAV Communications and Control Architecture

One critical result of these initial investigations was the development of a small UAV communications and control architecture.<sup>2</sup> In this architecture, low- and high-level control are implemented by two separate control modules. A third module provides wireless communications (Fig. 4).

The high-level controller acts as the user interface, accepting the mission definition from the user and delivering sensor data and mission status to the user. The supervisor also implements the high-level



**Figure 3.** (left to right) Preparation for launch, launch, and landing of the L3-BAI Aerosystems Javelin.

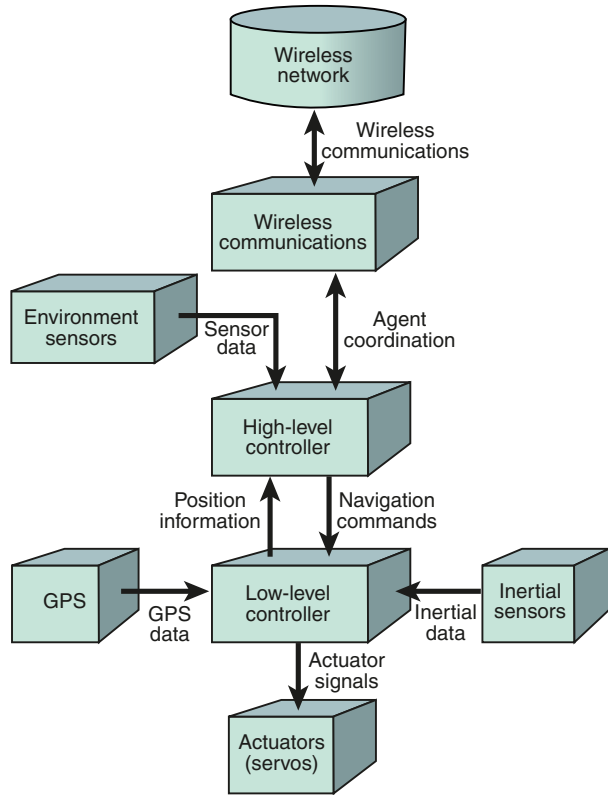


Figure 4. UAV system architecture block diagram.

coordination algorithms (implemented as FSA), as well as controls communications. The FSA output either course commands (i.e., “turn right”, “turn left”, “ascend”, “descend”) or waypoint commands (i.e., “go to this 3D point in space”), the latter being more typical. These commands are sent to the low-level controller—in this case an autopilot—which autonomously pilots the aircraft by moving surface controls (ailerons, rudders, flaps, elevators, etc.) and throttle based on inertial data from the onboard inertial sensors and position data from the onboard GPS. Commercial autopilots typically use a simple proportional-integral-derivative (PID) controller to map the sensor outputs to actuator motion.

Essential to all multivehicle coordination is wireless communications among swarm members. The supervisor is responsible for all two-way communications interface, content, and management functions.

Note that this architecture results in a closed-loop UAV system that can operate without communications to any ground controller. This is an important distinction from the architecture commonly used by other researchers of UAV swarming behaviors, in which the autonomy algorithms are run on a laptop and navigation commands are sent to the vehicle from a dedicated autopilot ground station.<sup>3-5</sup> Operationally, this latter method requires a communications link between the ground station and all UAVs throughout the entire mission at all times. The APL strategy allows the swarm to

operate completely autonomously beyond any communications link with the ground. Thus, both individual vehicles and vehicle swarms can operate completely independently of ground communications. This enables missions where the vehicles are beyond the reach of radio links and allows the vehicles to conduct their mission in areas where their radio link to the ground station is jammed.

### APL UAV Test Bed

As a follow-on to this effort, APL began development of its own UAV test bed. This activity was motivated by a desire for more flexibility in developing unique payload integration, vehicle control, and communications schemes.

APL worked with small-UAV pioneer Maynard Hill to develop two long-endurance air vehicles for future UAV investigations (Fig. 5). In addition to the Trans-Atlantic Model (TAM) vehicles developed by Mr. Hill, the APL UAV test bed has expanded to include two former Mig-117B target drones, which were provided to

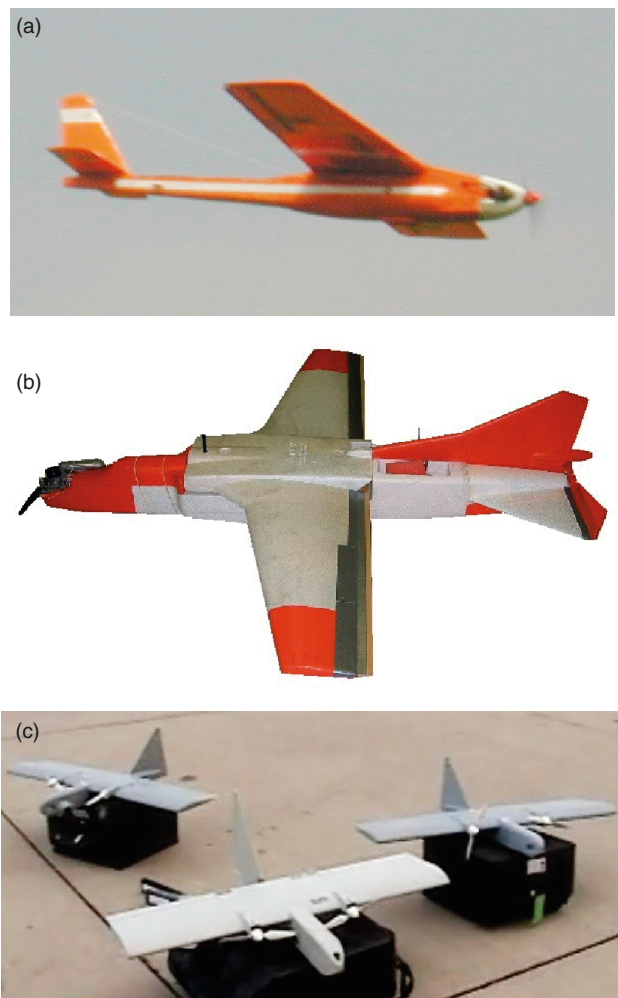


Figure 5. (a) TAM, (b) Mig, (c) DragonEye.

APL by the Army Research Laboratory (ARL) in Aberdeen, Maryland.<sup>6</sup> Two DragonEye variants have also been used as part of the APL flight demonstrations, but were given only temporarily by the Naval Research Laboratory (NRL) and are not part of the permanent APL capability. Because the DragonEye has been used by the Marines to such success recently, it is expected that the APL test bed will include those vehicles as a permanent resource sometime in the near future.

For onboard implementation of the autonomous swarming behavior FSAs, a small, lightweight, COTS single-board computer is used. A COTS IEEE 802.11b wireless local area network (WLAN) card is currently being used for swarm communications. For low-level vehicle control, two commercial autopilots have been used as part of the APL test bed: the Piccolo autopilot from Cloud Cap Technology and the MP2028 from MicroPilot.

## CONSENSUS VARIABLES APPROACH

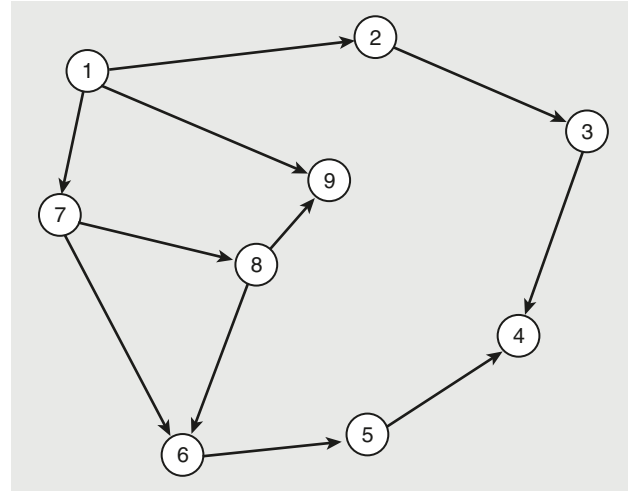
Another approach to autonomous, cooperative multiple vehicle behaviors was motivated by recent work on multi-agent coordination. This work is based on the idea of a consensus variable that is shared among the agents.<sup>7</sup> The minimal amount of information required for coordinated behaviors is assumed to be encapsulated in a time-varying vector called the coordination, or consensus, variable. Each agent carries its own local value of the consensus variable and updates that value based on the value held by other agents with whom the agent can communicate. Through proper definition of the consensus variable, and specification of rules for updating the value of this variable, it is possible for the value of the consensus variable to converge among the communicating agents.<sup>8</sup>

Assuming  $N$  agents with a shared consensus variable  $\xi$ , each agent has a local value of the variable given as  $\xi_i$ . Each agent updates its value based on the values of the agents with whom it can communicate. The continuous-time update rule is used, i.e.,

$$\dot{\xi}_i = - \sum_{N_i} k_{ij} (\xi_i - \xi_j),$$

where  $k_{ij} \geq 0$  is a weighting factor or gain that is non-zero when information flows from agent  $j$  to agent  $i$  and is zero otherwise. Under certain conditions, convergence of the form  $\xi_i \rightarrow \xi^*$  can be assured, with the final value of the consensus variable a function of both the initial conditions and the value of the gains.

For this to be true, the communication topology must form a spanning tree, wherein every agent does not require direct communications with every other agent, but at least indirectly all swarm members are connected (Fig. 6).<sup>8</sup> This is a key point: Consensus variable



**Figure 6.** Spanning tree created when global communications is achieved through indirect connections, even if the agents cannot communicate with each other directly.

formulation does not require global communications. Sharing only the consensus variable from “neighbor to neighbor” is enough under assumptions of the existence of a spanning tree in the communication topology. Note, however, that for our proof-of-concept demonstration, global broadcast was assumed.

This information sharing can be exploited for distributed signal processing. Assume that the distributed signal processing problem can be expressed as an optimization problem with a global performance function  $f$ , such as  $x^* = \min f(x)$  for some vector  $x$ . If a local estimate of  $x$  is placed at each node of a sensor network and employs a consensus variable-like algorithm of the form

$$x_i = \min f(\mathbf{s}_i)$$

and

$$\dot{\xi}_i = - \sum_{N_i} k_{ij} (\xi_i - \xi_j) + (x_i, \mathbf{s}_i),$$

then under some appropriate assumptions it can be shown that  $\xi_i \rightarrow x_i \rightarrow x^*$ . That is, it is possible to compute some aspect of a problem at a remote node (e.g.,  $x_i$ ) where perhaps some specialized information resides, and then communicate that result to the other nodes using the consensus variable approach. This results in a distributed system that converges to the global solution.<sup>9</sup>

## Dynamic Surveillance Network

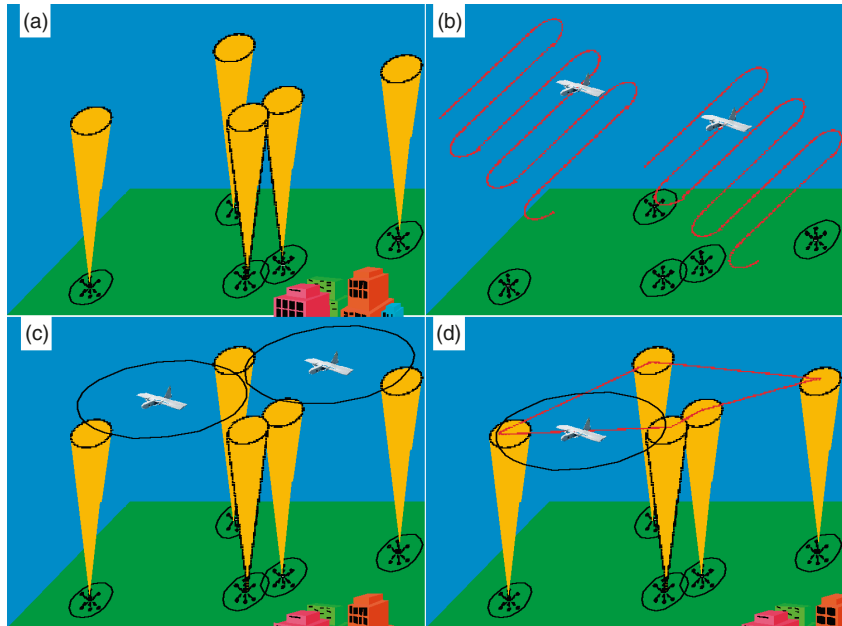
This consensus variables concept can be used for any number of multivehicle missions. APL developed a reference scenario called the Dynamic Surveillance Network (DSN),<sup>10</sup> which consists of a swarm of UAVs

exfiltrating data from a network of spatially distributed unattended ground sensors (UGSs) in an urban environment. This scenario was chosen since for military operations in urban environments, there is an increasing demand for distributed sensor networks to provide tactical or security forces timely, local, actionable operational intelligence. However, because of unreliable communications links in urban environments, and because these sensors may be in a denied area, direct communications back to a command post or to the warfighter cannot be assumed. Swarms of small, inexpensive UAVs can create robust ad hoc mobile air networks that provide this critical communications link. These swarms can be a fire-and-forget asset initiated by the soldier and can self-deploy and then operate continuously to reconfigure for optimal communications, adapting robustly to component failures and battle damage.

A proof-of-concept flight demonstration was developed in which multiple UAVs search, locate, and then coordinate with each other to most optimally exfiltrate the sensor data. Line-of-sight (LOS) communications were assumed from UGS to UAV, but not from UGS to UGS. Also, UGS-to-UAV communications were not assumed for the entire operational area. That is, a UAV could communicate with the UGS only within a limited area (an LOS cone above the sensor, typical of “urban canyons”). Finally, only an approximate knowledge of the UGS locations was assumed, necessitating an initial discovery process. The global system goal was to optimize sensor data exfiltration to the base station. It is during this process that the consensus variables approach is used, with optimal flight paths being the global variable.

The following describes the scenario CONOPS (concept of operations):

1. *Setup*: Sensors are deployed (Fig. 7a). It is assumed that in general the sensors cannot communicate with each other.
2. *Sensor discovery*: Before take-off, the user tasks the UAVs to find the sensors (Fig. 7b). Each UAV solves an optimal raster scan problem based on the known number of sensors and their approximate locations. The UAVs then negotiate to assign each UAV an individual flight pattern. Finally, the UAVs fly the fixed flight pattern and record the GPS location of the sensor nodes they have found.



**Figure 7.** Graphical depiction of a DSN scenario using two UAVs: (a) sensor deployment phase, (b) sensor discovery phase, (c) data exfiltration phase, and (d) adaptation phase.

3. *Data exfiltration* (collection and relay): After discovering the sensors, the UAVs solve an optimal coverage problem, determining the best course of action. Using the consensus variables approach, the UAVs then negotiate among themselves to assign each one an individual flight pattern based on the optimal solution (Fig. 7c).
4. *Adaptation*: The DSN system also adapts to changes in the network. If one of the sensor nodes becomes disabled, the UAVs detect this event and autonomously reconfigure. Adaptation also occurs if a UAV leaves the area. For example, periodically the UAVs agree that one of them should return to the base station and relay the data that have been collected from the sensors. When this occurs the remaining UAV reconfigures its flight pattern by resolving the optimal coverage problem assuming a new number of available resources (Fig. 7d). The initial flight pattern is resumed upon the return of the relay UAV.

### Flight Demonstration Hardware

Two different autonomous UAV platforms were used at various times during the flight testing of the DSN: the TAM and the Mig-117B target drone. Autonomous piloting of the vehicles was accomplished through the Piccolo autopilot. The consensus variables concepts were represented as a series of FSA, which generated DSN mission behaviors such as discovery, negotiation, data exfiltration, and adaptation (Fig. 8). A COTS, RISC-processor-based single-board computer was

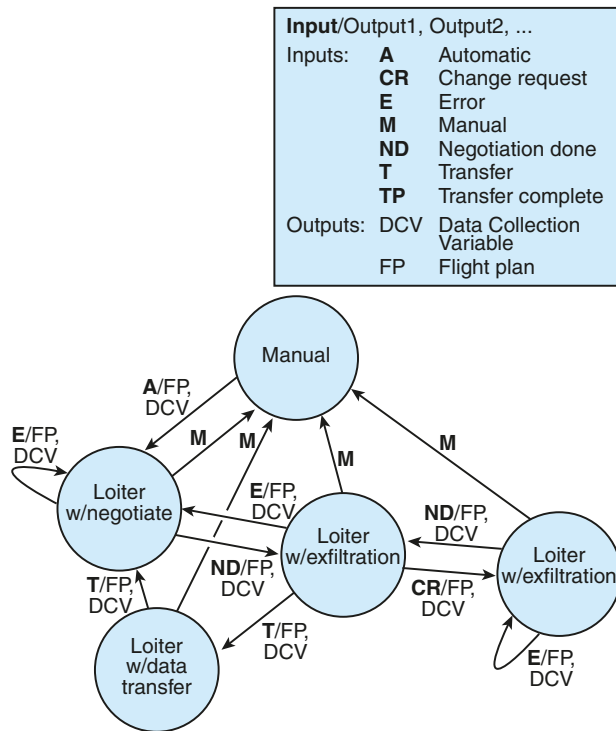


Figure 8. FSA used to implement DSN mission.

within the airspace and no connectivity among the sensors. Because the objective was to demonstrate the consensus variables concept and not the sensors, no sensor data were transferred over the wireless link. For the flight tests, the discovery, exfiltration, and adaptation stages used TCP/IP “pings” rather than actual data exfiltration as confirmation of UAV-to-UGS connectivity.

**Preliminary Demonstration Results**

Flight tests to date have demonstrated the ability to fly autonomously while communicating with the laptop-based sensor emulation stations. A flight pattern was programmed for a single UAV for an area that contained three emulated UGSs (Fig. 9). During the UGS flyovers, the UAV was able to ping the sensors, thus establishing sensor location. This preliminary test demonstrated the discovery phase of the scenario.

Testing was conducted by a joint APL/ARL team at various locations at Aberdeen Proving Ground (APG) in Aberdeen, Maryland. These flight demonstrations are ongoing. Future tests are planned for demonstrating two air vehicles coordinating to configure flight patterns that result in optimal connectivity.

used to implement these FSA, as well as enable user definition of the mission and control communications. As noted earlier, while the hardware suite and software algorithms differed completely from those used during the simple teaming approach flight tests, the basic UAV control and payload architecture was the same, as was the approach of using FSAs to implement behaviors.

The FSA that were used to implement behavior generated navigation commands (GPS waypoint updates) that directed the vehicle altitude, velocity, and course. These navigation commands were sent over a serial interface to the autopilot in flight. The onboard computer also provided the drivers, data interface, control, and power for the WLAN card.

The ground sensors used for this test were laptops outfitted with WLAN cards. To simulate communications in an urban canyon, highly directional upward-pointing antennas were used that provided limited zones of connectivity

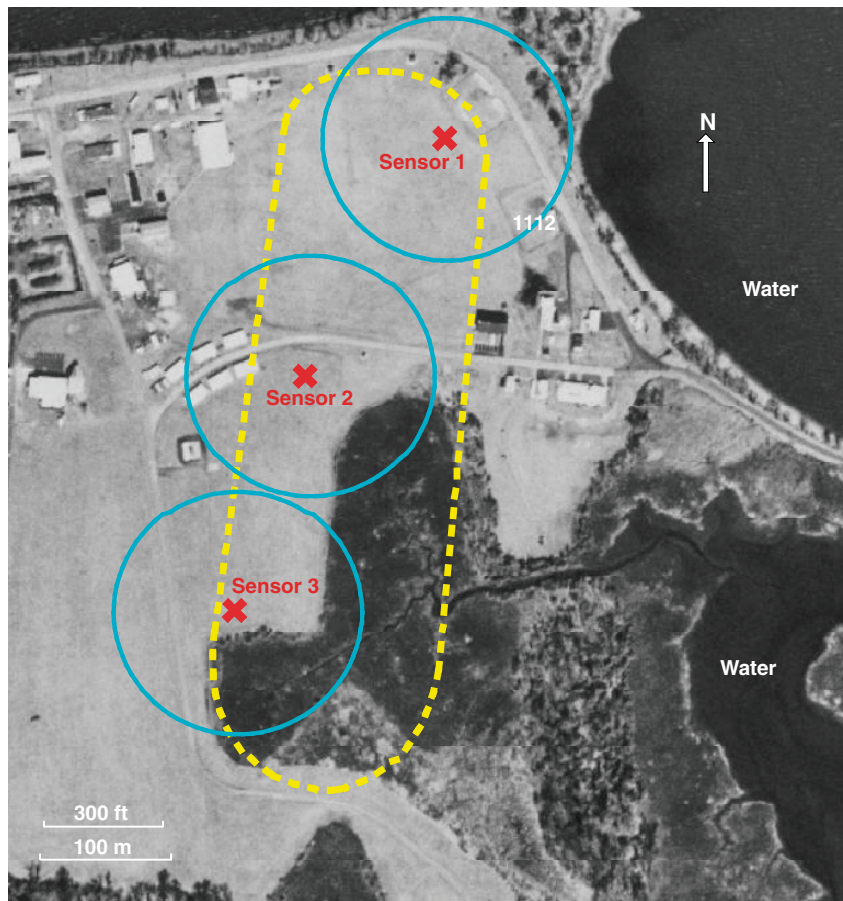


Figure 9. Demonstration of discovery phase.



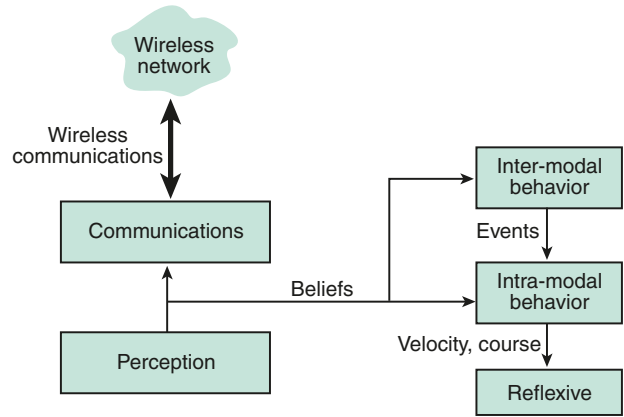
## STIGMERGIC POTENTIAL FIELDS APPROACH

Stigmergy represents a fundamentally different approach to UAV swarming than the consensus variables or simple teaming concepts. Those approaches coordinated vehicle actions through negotiation among swarm members or supervisory control designed to address mission objectives. With stigmergy, the cooperative problem is solved by heterarchically organizing agents that coordinate indirectly by altering the environment and reacting to the environment as they pass through it. That is, the agents do not negotiate directly with each other to mutually decide on courses of action. Stigmergy is accomplished through the use of locally executed control policies based on potential field formulas. These formulas, called stigmergic potential fields (SPFs), are used to coordinate movement, transient acts, and task allocation among cooperating vehicles.<sup>11</sup> While the previous approaches theoretically produce more optimal swarm behaviors, stigmergy results in behaviors that respond more quickly to changing events and can support swarms containing more members.

With SPFs, a virtual potential field is associated with all the germane entities contained within a locally held model of the vehicle's world. These models typically include peer vehicles and elements that are impacted by peer vehicles. These fields are used to influence vehicle action, most notably movement. The forces associated with the fields may be attractive (directing the vehicle toward a point), repulsive (directing the vehicle away from a point), or complex (a combination of attractive and repulsive fields) (Fig. 10). At any given time, the total force on a vehicle is the summation of all attractive, repulsive, and complex forces due to all known influences.<sup>12</sup> This approach to vehicle control borrows heavily from earlier work in vehicle path planning with potential fields. This work most closely relates to Arkin's motor schema behavior, Zambonelli's co-fields, Spears' physicomimetic behavior, and Koren's solutions to problems of local minima and undesirable oscillations.<sup>13-16</sup>

### Hierarchical Architecture and Communications Protocol

The hierarchical control architecture developed for the demonstration of these SPF concepts consists of five essential layers (Fig. 11). Each layer is an independently

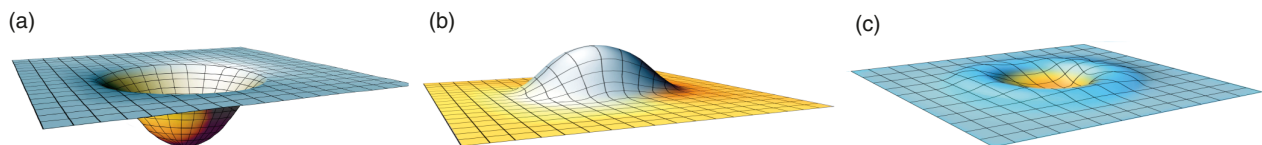


**Figure 11.** Hierarchical control architecture used in the implementation of the stigmergic approach.

executing control process. The upper layers exercise control over the lower layers by asynchronously altering the objectives of the lower levels. The top-most inter-modal behavior level controls transitions among the other behaviors. The intra-modal behavior level realizes specific behavioral functions such as regulation of transient actions (e.g., vehicle course and speed). The reflexive behavioral layer provides collision avoidance and maneuvering while pursuing the course ordered by the intra-modal layer.

The sharing of knowledge among swarm members represents a cornerstone of the SPF approach. "Beliefs" (data structures) represent a swarm member's knowledge of abstract objects in its environment, as well as its own position and state. These objects may be moving or stationary. Moving objects may include unknown vehicles, noncombatants, cooperating members of the swarm group, identifiable targets, and legitimate threats. In addition to those categories, stationary objects include topography obstacles.

Beliefs are generated by the perception level, which senses the environment and vehicle status using physical sensors. Beliefs are provided by the perception layer to the vehicle's inter-modal, intra-modal, and communications layers. The communications layer transfers these beliefs to other swarm members, as well as receives beliefs from other swarm members. The beliefs from the vehicle's own perception layer, along with the beliefs from other swarm members, represent the core input to the algorithms that generate the stigmergic potential fields.



**Figure 10.** Fields that are (a) attractive, (b) repulsive, and (c) complex. An attractive field results in the vehicle being drawn toward its center, a repulsive field results in the vehicle being drawn away from its center, and a complex field is the combination of attractive and repulsive forces in one location.

Clearly, the communications layer is a critical link in enabling stigmergic swarm behaviors. A communications framework was developed for belief transfer that uses a modular, multilayered architecture (Fig. 12).<sup>17</sup> This framework was designed to facilitate distributed collaboration over any mobile ad hoc network (MANET). At the core capabilities layer, the transmit subsystem provides periodic and on-demand message transmit capabilities that can be accessed by any vehicle software application.

The framework provides several functions, including a neighbor discovery capability that allows any node to identify peer nodes within its immediate wireless coverage. It also allows any application to detect swarm size, as well as join and leave events. These events are commonly encountered in wireless ad hoc networks where rapidly changing topology, varying RF link quality, and unpredictable obstacles can cause partitioning and merging of swarm clusters. The framework capabilities include collection, logging, and broadcast of various communication statistics.

This framework was designed to be highly adaptable for use in various applications and deployments where impromptu ad hoc communications are required. The system allows activation/deactivation of any of its subsystems at software construction time and at run-time using configuration properties and the service interface. In most deployments scenarios, manual setup and administration are not required to establish the wireless network. Default configuration properties can

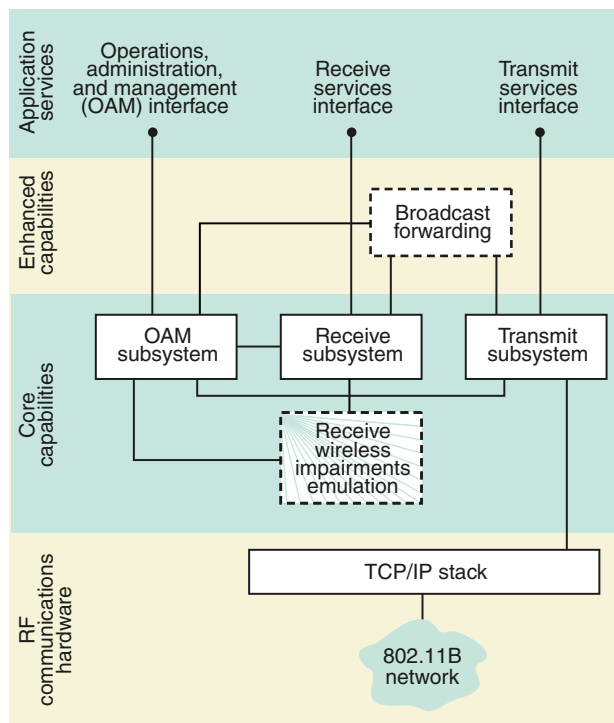
be established to allow operations in a wide range of environments. The framework uses these defaults to automatically initialize the communications software framework, without requiring the high-level application to understand the internal configuration details.

The framework protocols were designed to limit computational and communication resources, thereby limiting energy consumption. For example, forwarding specific application messages across one or more hops is allowed when these messages have been marked by the sending node for multi-hop forwarding. This forwarding service is designed to eliminate broadcast storms that occur with simple forwarding flooding techniques. The forwarding service protocol does not rebroadcast a message unless this rebroadcast allows the message to reach new receivers.

### Demonstration of Communications Protocol

An extensive series of simulations, HIL tests, and flight tests were conducted to validate the communications protocol. The flight tests were conducted at APG by a team from APL and ARL. The objective was to use an airborne WLAN node to hop messages between ground assets that could not communicate directly. The ground assets consisted of an unmanned ground vehicle (UGV), two UGVs emulated on a single laptop, and a laptop ground station that displayed messages and tracked vehicle movement. A helicopter UAV equipped with a WLAN card acted as the airborne node.

The three ground nodes were distributed throughout the test field such that none of the nodes had direct communications with each other. This was accomplished by locating nodes far apart or behind obstacles such as buildings. The primary test metric was the successful multicast hopping of packets between ground nodes. This hopping was substantiated by using the ground station's visual display of message transfers and vehicle motion. With the air vehicle at an altitude of approximately 100 m, it was flown to several locations around the test field. Within minutes the UAV reached a location that allowed messages to be forwarded from the robot UGV to the ground station, and from one of the emulated UGVs to the ground station. The laptop could emulate only one UGV at a time. When emulation of the first UGV was terminated and emulation of the second UGV was initiated, this was also reflected on the ground station. As a final test, the second emulated UGV attempted to ping the ground station but was unsuccessful despite messages being hopped from that UGV to the ground station. This verified that the emulated UGV was truly not in direct communications with the ground station and the airborne node was really hopping selective application messages as opposed to acting like an infrastructure relay node.



**Figure 12.** UAV swarm communications modular, multilayered architecture.

## Demonstration of the SPF Approach

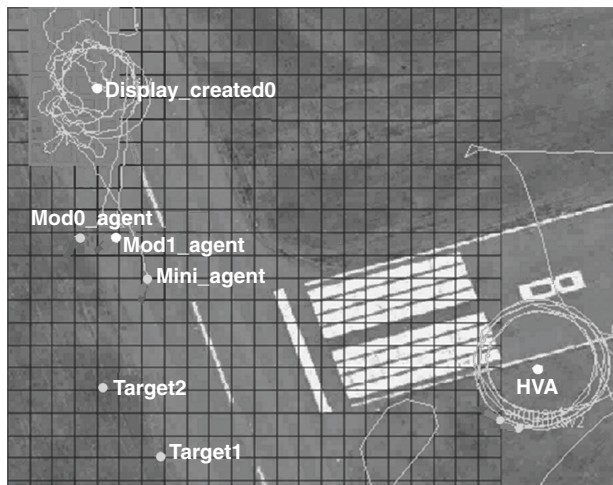
The SPF approach was used to implement seven behaviors useful for military operations: search, tracking, classification, friendly influence, circular formation, linear formation, and pheromone-based search. One of these behaviors, the cooperative search behavior, is described in some detail below.

The goal of cooperative search is to direct vehicle motion to accomplish complete sensor coverage of an area designated by a search order. Simple cooperative search is determined at any arbitrary time  $t$  by having each vehicle apply the field

$$g_p(r) = \frac{[1 - s(1 - \lambda)^{t - t_s}]}{(1 + r)^k},$$

where  $\lambda \propto v_{obj}$  to all locations in  $Z^2$  space. The coefficient  $s$  is the probability of detection for a sensor observing the point in question, and  $t_s$  is the most recent time the point was surveyed. The decay in knowledge of a previously observed area is controlled by the decay constant  $\lambda$ , which is proportional to the expected speed of an adversary  $v_{obj}$ . The constant  $k$  is used to adjust the cohesiveness of a search pattern. Further improvements to the cooperative search may be found by applying secondary fields to cooperating sensor and communications platforms.

From July to September 2004 a series of experiments was conducted to demonstrate SPFs in a real-world scenario. The behaviors demonstrated included search, tracking, classification, friendly influence, circular formation, and linear formation (Fig. 13). These experiments established the ability of a small heterogeneous swarm (two UAVs and four UGVs) to achieve a series of objectives provided asynchronously by three uncoordinated operators exercising effects-based control. The demonstrations increased in complexity throughout



**Figure 13.** UGVs classify contacts found by UAVs escorting a convoy.

the test period, culminating in an experiment to help protect fixed and mobile assets in an obstacle-laden urban area.

The UAV/UGV swarm was responsible for providing reconnaissance, surveillance, and target acquisition capabilities in support of the mission objectives. Mig-117B drones were used as the UAV vehicles, while the MicroPilot autopilot was used to control UAV flight. The FSAs that implemented the stigmergy behaviors and communications protocols were run on a laptop and communicated to the UAV through the MicroPilot ground station. In other words, the FSAs were not implemented on an onboard computer. Note, however, that this was *only* because the MicroPilot autopilot version that was used did not support commands from any source other than its ground station. In other words, the FSAs were not so computational intensive that they would not have fit on a small, single-board computer. Future migration to other autopilot systems will enable a completely integrated onboard command and control architecture.

These tests were also conducted at APG by a joint APL/ARL team. To ensure that the most realistic environmental stimuli possible were provided to the vehicle sensors within the limited test area, significant engineering of the environment was necessary. Runways were used to represent a road network. An urban area, including a power station and several buildings, was constructed of wood-reinforced foam board. A convoy was represented by an instrumented van containing an operator's station. Unknown contacts were represented by test personnel outfitted with range-limited position beacons.

The scenario was instantiated by an operator who broadcast a belief that a fixed asset (a notional power station at a known location) required protection. The autonomous UGVs, which had been prestaged  $\approx 90$  m away, responded by searching the urban area. Once the area was completely searched, the UGVs secured the power station within a moving circular perimeter. Concurrently, the local road network was patrolled by autonomous UAVs.

After several minutes, a second operator located within a mobile asset (a notional convoy) broadcast the belief that the convoy required continuous protection. When the UAVs detected the convoy they shifted behavior, changing from patrol to escort, providing air support for the convoy as it moved down the road. Several contacts were staged at a road intersection. The contacts were identified by range-limited beacons designed to simulate an IR detection capability. As the convoy moved along the road network, the escorting UAVs detected these unknown contacts at an intersection. The existence of these unknown contacts was broadcast to all vehicles via the wireless network. The UGVs reacted to these new beliefs by forming a linear barrier between the unknown contacts and the power station. One UGV, which was equipped with a directional acoustic sensor, shifted to

a classify behavior, closing with the unknown contacts including both noncombatants and a hostile target. The target was distinguished by an audible tone. The UGV with the classification sensor approached the contacts, classifying each as hostile or friendly strictly on the basis of the presence or absence of the tone. Once the hostile contact was identified, all the UGVs responded by transitioning to track behavior. The hostile contact was pursued, surrounded, and neutralized by the UGVs.

The convoy-based operator then broadcast a belief that the intersection may contain explosives. Simultaneously, the original operator reenforced the belief that the power plant required protection. The UGVs responded by self-organizing to accomplish both tasks. Significantly, the actions executed by the UGVs to this point were in direct response to the stimulus provided, with no human intervention.

The operational scenario was repeated successfully many times. While the specific movement of individual vehicles varied greatly, the collective behavior and effectiveness remained constant. Additional complexity was introduced by adding software agents such as virtual UAVs, an actual UGS, and mixes of virtual and actual UGVs.

Hardware failures played a key role in understanding and demonstrating the robustness of the control algorithms. Two UGVs failed: one lost mobility and the other experienced a total power failure. The remaining vehicles continued to perform the mission with no explicit replanning and no operator intervention. Despite the hardware failures, the remaining vehicles were able to complete the mission objectives with only a 15% increase in scenario completion time.

Experiment metrics included repeatability, consistency, behavioral stability, and robustness to failure of the distributed multivehicle control system. Extensive recordings were made of vehicle positions, behavior transitions, position histories, and scenario events. The results were promising. The circular and linear formation behaviors, used for fixed and mobile asset protection, provided very stable and consistent performance. The consistency of the search and track behaviors, however, was more varied. This was consistent with earlier simulated results that showed search efficiency to depend on the density of the vehicle community, size and dispersion of the search, speed of the vehicle, and sweep rate of the sensor. Since these dependencies are largely observable by the vehicles, it is believed that adaptive fields may be engineered to provide a consistently high degree of efficiency.

### Application of the SPF Approach for Chemical/Biological Plume Detection

In September 2005 the SPF approach was applied to a hazardous plume detection concept of operations.

Rather than intended as a demonstration of SPF, this was an actual application of SPF to a defined problem.

The test was conducted at Dugway Proving Ground (DPG) in Dugway, Utah, by a joint APL/NRL test team. A modified version of the DragonEye air vehicle was used, along with a proprietary NRL autopilot. The FSAs that controlled the vehicle behavior were implemented on a laptop PC, which was connected to the autopilot ground station. To control velocity, altitude, and course, joystick commands spoofed by an APL-developed software interface layer were sent to the ground station. As with the APG demonstrations of the SPF approach, this suboptimal hardware architecture was driven by limitations of the autopilot rather than the computational requirements of the software.

For this mission, a standoff detector or human observer provided to the UAV only notional information about the location of a chemical/biological release (Fig. 14). Without user input (only launching), the UAV was deployed to the plume and conducted an autonomous search of the designated area (Fig. 15). Simulated UAV chemical/biological sensors autonomously analyzed the suspected release zone. In the case of the chemical plume, the sensors identified the chemical agent and characterized the spatial extent of the plume (Fig. 16). In the case of the biological plume, the sensors collected the biological agent to be analyzed post-landing. Because

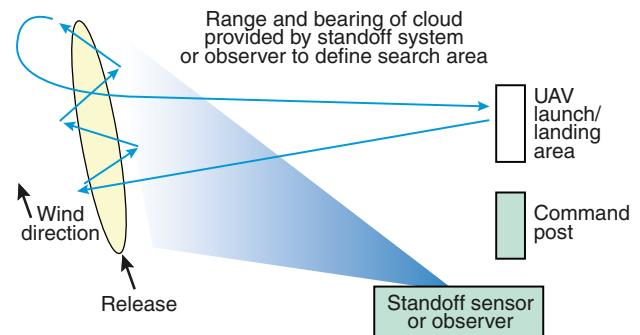


Figure 14. Chemical/biological plume detection scenario.

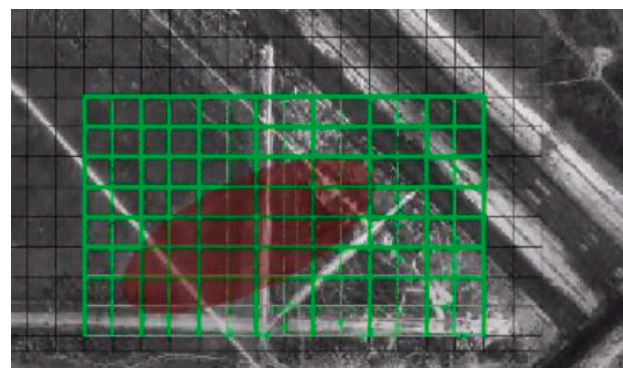
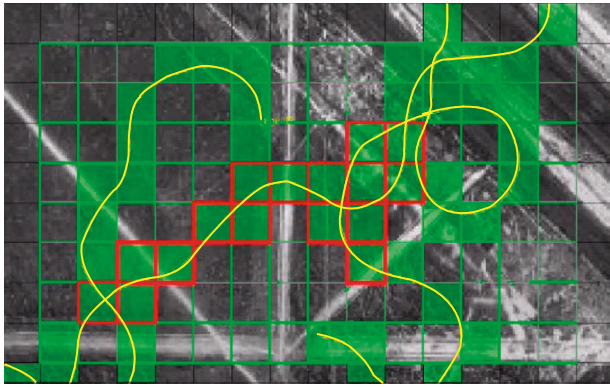


Figure 15. Search area defined *a priori* by green blocks; simulated plume shown in red.



**Figure 16.** Shaded green blocks indicate area searched by the UAV, red outlined blocks indicate blocks within which the UAV detected the simulated plume, and the yellow line indicates vehicle ground track.

the objective of this mission was to test autonomy, the hazardous plume was simulated in software. That is, detection was of a virtual plume using virtual sensors rather than an actual plume using physical sensors.

Because of vehicle malfunctions, the DPG tests were conducted with only a single UAV, though multiple vehicles are planned for future tests. Nonetheless, it provided validation of the SPF approach to real-world problems.

## AUTONOMOUS VEHICLE CONTROL

Just as the flight tests represented the first step in bringing APL's UAV swarm methodologies to operational fruition, they also exposed critical deficiencies in the capability and functionality of commercial small-UAV autopilots. For instance, COTS autopilots typically do not support the high-performance maneuvers required for urban operations, nor do they offer adequate compensation for high winds. COTS autopilots tend to be designed for preprogrammed waypoint navigation entered by a user through a dedicated ground station rather than dynamic programming and reprogramming over some onboard interface from an onboard computer. Finally, the COTS autopilots designed for this class of air vehicle are based on PID control. In addition to the limitations of PID controllers, they also require labor- and time-intensive procedures to establish the PID gains. Not only are these gains different for every UAV type, but recalibration of the gains is often required whenever a UAV is modified (e.g., new sensors are added that change the center of gravity or aerodynamics). This has implications both for permitting field modifications and for compensation of in-flight battle damage.

APL is currently engaged in an internal development program to create a robust second-generation UAV control architecture that includes an advanced autopilot and a flexible supervisory control software architecture.

Closely coupled with the autopilot will be a reflexive controller that takes priority control of the vehicle for purposes of collision avoidance, formation flying, and terrain following. An arbitrator block is used to arbitrate autopilot and reflexive control signals. Together, these modules form an advanced controller that will allow platoon-level Class I UAVs to more effectively support the complex missions enabled by innovative swarming techniques (Fig. 17).

Several alternatives to traditional PID controllers have been offered for low-level autopilot control, including scheduled PID, hybrid fuzzy PID, dynamic surface control, biomorphic control, and model predictive control (MPC). In the current effort, APL is working jointly with researchers at Earthly Dynamics to investigate MPC as the control methodology for an autopilot. Although not intelligently "adaptive," this type of control has been shown to be forgiving of radical changes in vehicle configuration and aerodynamics (e.g., from damage to wing during flight). MPC also has a demonstrated capability of maintaining vehicle course in high winds. User development of the MPC coefficients for each different airframe and configuration is usually straightforward, often not even requiring a flight component. Also, in addition to implementing a more advanced control methodology, this autopilot will contain hooks to allow seamless two-way interface with the onboard supervisory control. The culmination of this investigation will be a rigorous series of flight tests of an MPC-based autopilot on a small UAV. These tests will demonstrate high-performance maneuvers in a simulated urban environment and reliable course tracking in a high-wind regime. Another product of this effort will be a portable laptop PC-based ground station that, unlike most COTS autopilots, will require no additional hardware other than a plug-in WLAN card with external antenna.

## CONCLUSIONS

The vision for the future battlespace includes both manned and unmanned systems operating jointly for a common objective. Multiple UAVs operating cooperatively, in tandem, expand greatly on the capability of individual vehicles. However, the full utility of multi-UAV systems will only be realized if they can operate autonomously: able to fly, adapt, communicate, negotiate, and carry out missions with no human in the loop.

APL has developed, and demonstrated in the field, several approaches to cooperative autonomy. Both the development and demonstrations of the consensus variables and SPF approaches continue. (The simple teaming approach is constrained by its ability to support very complex missions and high numbers of vehicles.) These activities have also resulted in the establishment of robust payload architectures, an efficient communications framework, a systems integration process for small

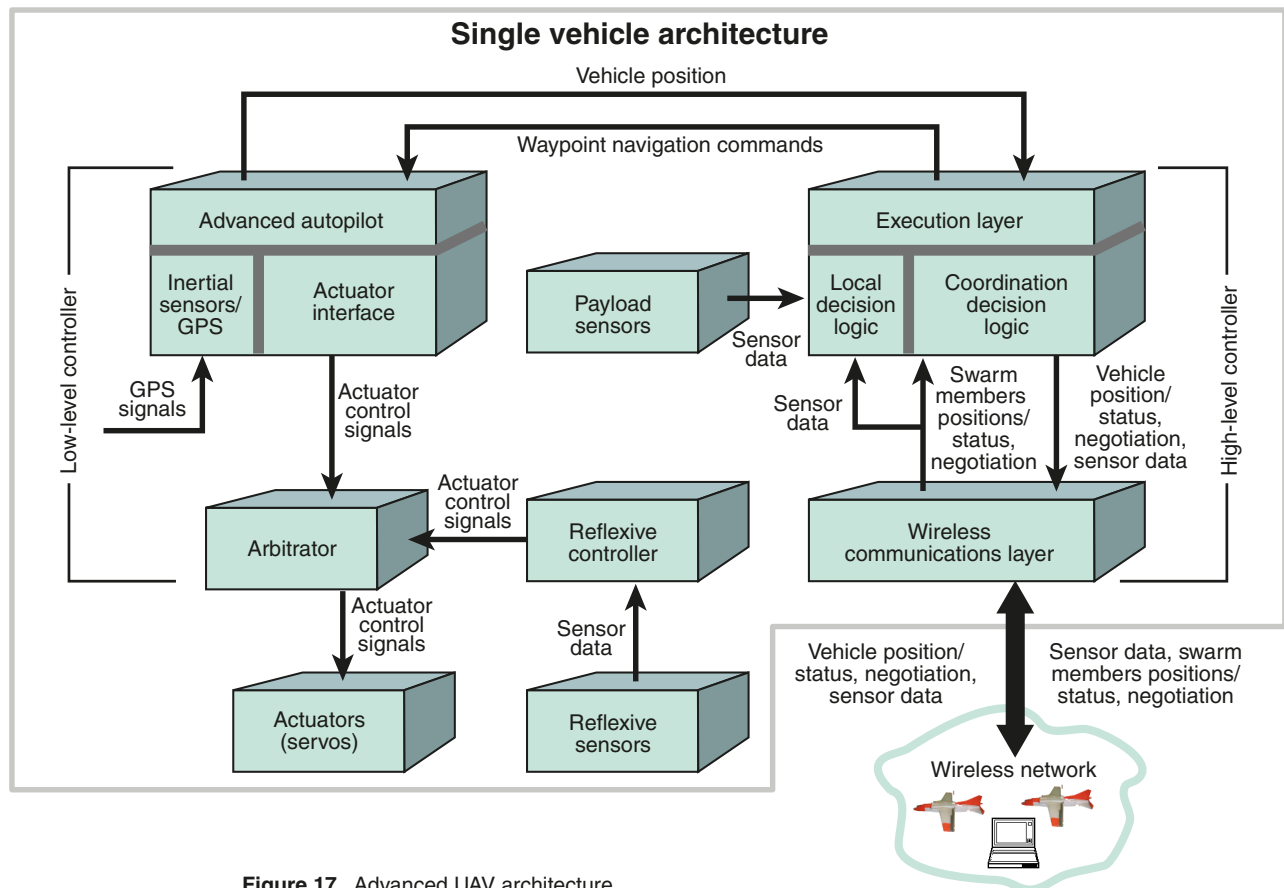


Figure 17. Advanced UAV architecture.

vehicles, and an APL UAV test bed. Future developments will result in an integrated, advanced UAV control architecture that will be realized in hardware.

With interest in both small systems and autonomy growing rapidly throughout the national defense communities, the future for multivehicle swarming concepts using small, autonomous UAVs looks bright. As a result of its commitment to developing these concepts, APL is positioning itself to be a key innovator in these technologies.

**ACKNOWLEDGMENTS:** The UAV activities described in this article span several projects that include contributors from across APL as well as outside APL. The initial UAV teaming activities were led by Hal Gilreath, and key contributors included George Barrett, Jim Bruzek, Dale Elkiss, Timothy Gion, Kenneth Grossman, and Frank Weiskopf. BAI Aerosystems also must be recognized for providing vehicles and flight services. Establishment of the initial APL UAV test bed relied heavily on the experience and knowledge of Maynard Hill, who was assisted by APL student intern Cyrus Abdollahi. Implementation and demonstration of the stigmergic fields approach using heterogeneous unmanned vehicle teams were made possible by the considerable efforts of Bob Chalmers, Chris Chiu, Chad Hawthorne,

Todd Neighoff, and Jason Stipes. Osama Farrag created the communications framework that made realization of those concepts possible. Dennis Lucarelli contributed to solving the DSN problem using consensus variables, and implementation and testing of the DSN concept would not have been possible without the efforts of Michael J. White. ARL personnel have played key roles throughout many of the flight demonstrations, particularly Harris Edge, Robert Hayes, and Stephen Wilkerson. The Dugway test team was led by Eric Van Gieson, and included Mr. Chalmers, Mr. Chiu, Peter Fuechsel, Mr. Hawthorne, and Ronald Mitnick. The advanced UAV controller is being developed with Earthly Dynamics founder Mark Costello, a professor at Georgia Tech University, and senior partner Nathan Slegers, a professor at the University of Alabama at Huntsville. Additional internal collaborators include Andrew Harris, Tony Le, Jay Moore, Neil Palumbo, and Matthew Para.

#### REFERENCES

- <sup>1</sup>Weiskopf, F., Gion, T., Elkiss, D., Gilreath, H., Bruzek, J., et al., "Control of Cooperative Autonomous Unmanned Aerial Vehicles," in *Proc. AIAA 1st Technical Conf. and Workshop on Unmanned Aerospace Vehicles, Systems, Technologies, and Operations*, AIAA 2002-3444 (2002).
- <sup>2</sup>Bamberger, R., Bruzek, J., Gilreath, H., Gion, T., and Grossman, K., "An Architecture for Multi-Vehicle Autonomy with Small UAVs," in *Proc. AHSI 59th Annual Forum*, Phoenix, AZ (2003).

- <sup>3</sup>Kaminer, I. I., Yakimenko, O. A., Dobrokhodov, V. N., Lizarraga, M. I., and Pascoal, A. M., "Cooperative Control of Small UAVs for Naval Applications," in *Proc. 43rd IEEE Conf. on Decision and Control*, pp. 626–631 (2004).
- <sup>4</sup>King, E., Kuwata, Y., Alighanbari, M., Bertuccelli, L., and How, J., "Coordination and Control Experiments on a Multi-vehicle Testbed," in *Proc. 2004 American Control Conf.*, pp. 5315–5320, (2004).
- <sup>5</sup>Sward, R. E., "Proving Correctness of Unmanned Aerial Vehicle Cooperative Software," in *Proc. 2005 IEEE Networking, Sensing and Control Conf.*, pp. 767–771 (2005).
- <sup>6</sup>Moore, K. L., White, M. J., Bamberger, R. J., and Watson, D. P., "Data Exfiltration from Unattended Ground Sensors Using Cooperating UAVs," in *Proc. SPIE Security and Defense Symp., Unmanned Ground Vehicles VII*, Orlando, FL, (2005).
- <sup>7</sup>Ren, W., Beard, R. W., and McLain, T. W., "Coordination Variables and Consensus Building in Multiple Vehicle Systems," in *Proc. Block Island Workshop on Cooperative Control*, Springer-Verlag Series: *Lecture Notes in Control and Information Sciences* (2003).
- <sup>8</sup>Moore, K. L., and Lucarelli, D., "Forced and Constrained Consensus Among Cooperating Agents," in *Proc. 2005 IEEE Int. Conf. on Networking, Sensing, and Control*, Tucson, AZ (2005).
- <sup>9</sup>Moore, K. L., and Lucarelli, D., "Consensus Variable Approach to Decentralized Adaptive Scheduling," in *Proc. 5th Int. Conf. on Cooperative Control and Optimization*, Gainesville, FL (2005).
- <sup>10</sup>Moore, K. L., White, M. J., Bamberger, R. J., and Watson, D. P., "Cooperative UAVs for Remote Data Collection and Relay," in *Proc. AUUSI Unmanned Systems North America 2005*, Baltimore, MD (2005).
- <sup>11</sup>Scheidt, D. H., Stipes, J. A., and Neighoff, T. M., "Cooperating Unmanned Vehicles," in *Proc. IEEE Conf. on Networking, Sensing and Control*, Tucson, AZ (2005).
- <sup>12</sup>Chalmers, R. W., Scheidt, D. H., Neighoff, T. M., Witwicki, S., and Bamberger, R. J., "Cooperating Unmanned Vehicles," in *Proc. AIAA Intelligent Systems Technical Conf.*, AIAA-2004-6252, Chicago, IL (2004).
- <sup>13</sup>Arkin, R., *Behavior-Based Robotics*, MIT Press (1998).
- <sup>14</sup>Spears, W. M., and Gordon, D. F., "Using Artificial Physics to Control Agents," in *Proc. IEEE Int. Conf. on Information, Intelligence and Systems* (1999).
- <sup>15</sup>Mamei, M., Zambonelli, F., and Leonardi, L., "Co-Fields: A Unifying Approach to Swarm Intelligence," in *Proc. 3rd Int. Workshop on Engineering Societies in the Agents' World*, Madrid (2002).
- <sup>16</sup>Koren, Y., and Borenstein, J., "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," in *Proc. IEEE Conf. on Robotics and Automation*, Sacramento, CA, pp. 1398–1404, (1991).
- <sup>17</sup>Bamberger, R., Scheidt, D., Hawthorne, C., Farrag, O., and White, M., "Wireless Network Communications Architecture for Swarms of Small UAVs," in *Proc. AIAA Unmanned Unlimited Conf.*, AIAA-2004-6594, Chicago, IL (2004).

## THE AUTHORS



Robert J. Bamberger Jr.



David H. Scheidt



David P. Watson



Kevin L. Moore

**Robert J. Bamberger Jr.** is the Principal Investigator (PI) of the Advanced Autopilot IR&D project and has supported all three of APL's major UAV swarm projects. He is a Senior Professional Staff member in the Research and Technology Development Center (RTDC). **David P. Watson** and **David H. Scheidt** jointly lead APL's Science and Technology thrust in autonomous systems. Mr. Watson has performed research and development in autonomous systems for underwater, air, and space applications. He is the Supervisor of the Systems and Information Sciences Group in the RTDC. Mr. Scheidt was the PI for the investigation of stigmergic fields strategies for multiple UAV cooperation and has led numerous efforts in distributed intelligent control systems, both at APL and in industry. He is a member of the APL Principal Professional Staff in the National Security Technology Department. **Kevin L. Moore** was the PI for the investigation of consensus variables as an approach for multivehicle negotiation. When at APL, Dr. Moore was a senior scientist in the RTDC. He is currently the G.A. Dobelman Distinguished Chair and Professor of Engineering in the Division of Engineering at the Colorado School of Mines. Further information on autonomy research at APL can be obtained from Mr. Watson. His e-mail address is david.watson@jhuapl.edu.