Data exfiltration from unattended ground sensors using cooperating UAVs

Kevin L. Moore[1], Michael J. White, Robert J. Bamberger, and David P. Watson
Research and Technology Development Center
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723-6099

ABSTRACT

Advances in networking, communications, and sensor technology have made it possible to deploy networks of spatially-distributed sensors to provide tactical or security forces with critical operational intelligence. Many application arenas have requirements for low-power, long-life sensors and stand-off for operations personnel, dictating the need for so-called unattended ground sensor networks (UGS). However, often the environment (e.g., an urban environment) limits communication capabilities. To solve this problem, in this paper we propose the use of multiple, cooperating unmanned aerial vehicles (UAVs) to collect, or exfiltrate, the data from the UGS. In our scenario several prototype sensor clusters are deployed in relatively unknown locations, with a limited communication range. Each cluster's dedicated base/bridge node communicates via an IEEE 802.11b wireless link. The UAVs execute a cooperative flight pattern to first find the sensor clusters, to then collect their data, and to relay that data to a ground station. Throughout the process, the UAVs cooperate autonomously to achieve the goal, reconfiguring as needed in response to changes in the mission, in the targets (sensors), or in the available resources (the UAVs themselves). In the paper we describe our approach to cooperative behavior, our aircraft, and some preliminary experimental results.

Keywords: Sensor networks, intelligent control, UAVs, surveillance, mobile ad-hoc networks, robotics, command and control, architecture, resource allocation, data exfiltration.

1. INTRODUCTION

As military operations become increasingly focused on networked local units in urban environments there will be an increasing demand for these networks to provide timely local, actionable surveillance information that is unavailable from traditional ISR platforms. One solution to this need relies on advances in networking, communications, and sensor technology, which have led to the possibility of deploying networks of spatially-distributed sensors that can provide tactical or security forces with critical operational intelligence. For example, we can posit the idea of formations of small, inexpensive UAVs to create robust ad-hoc mobile air networks for battlefield situational awareness. Such formations can be a soldier-initiated, fire-and-forget asset that is able to self-deploy, configure and then operate continuously to respond to changing quality of service (QoS) requirements from ground-based tactical user nodes. Notionally these formations would be able to respond robustly to component failures and battle damage through dynamic, autonomous reconfiguration, while altering relative ranges and network topology to maintain connectivity and QoS. Another scenario of particular interest is distributed sensor networks for covert and other surveillance and monitoring activities. For instance, a special operations team might furtively deploy a distributed sensor network to detect, possibly identify, and record vehicle movement in and out of a building complex of interest. Monitoring the data collected by such a network could, for example, provide key information about patterns of activity in the building complex that would be needed to support a planned covert operation. Both examples given above share a number of common features, including:

- Some type of information of interest (e.g., battlefield conditions, vehicle movements around a building cluster).
- A means of acquiring and possibly interpreting this information locally (e.g., tactical units on the battlefield, vehicle detection sensors).

[1] Send correspondence to K. L. Moore at kevin.moore@jhuapl.edu.

Unmanned Ground Vehicle Technology VII, edited by Grant R. Gerhart, Charles M. Shoemaker, Douglas W. Gage, Proceedings of SPIE Vol. 5804 (SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.604391

- A means of exfiltrating or communicating this information to interested decision-makers (e.g., UAVs in both examples).
- The possible need for low power sensors.
- The fact that both data acquisition sensors and communication nodes might be mobile.
- The fact that global and local communications might not be ensured.

This last point is particular important and implies that data exfiltration will sometimes only be able to provide sporadic data collection. For instance, in an "urban canyon" the presence of buildings and other structures that cause communication dead zones, together with the need for low-power, long-duration surveillance, may lead to a situation where data can only be exfiltrated intermittently and in close proximity to the sensor.

The example scenarios and their characteristics lead to the general notion of a dynamic surveillance network (DSN), illustrated in Figure 1, which shows a cooperating system of spatially-distributed nodes, including UAV sensors (UAV-s), unmanned ground vehicle sensors (UGV-s) and unattended ground sensors (UGS). The network could also include what we call unattended ground vehicle actuators (UAV-a), unattended ground actuators (UGA), and unattended air vehicle actuators (UAV-a). In general a node can be a sensor, an actuator, a communication node, or a combined, co-located sensing, actuation, and communication node. A communication node could be short-range or long-range, depending on its power availability. Likewise, a sensor node might in fact be a short-range network, or sensor cluster, with limited-power, low-range communication capabilities. The idea is that cooperating groups of such nodes can be deployed to work together to achieve some type of global mission or objective. In this paper we present preliminary results of research aimed at demonstrating the use of such cooperative behavior, applied to a specific application: a small-scale dynamic surveillance network for the problem of UAV-based data exfiltration from low-power unattended ground sensors in an urban environment.

The remainder of the paper is organized as follows. In the next section we describe the DSN scenario in more detail. This is followed by an overview of our algorithmic approach to multi-vehicle cooperation. We then describe our hardware and its architecture and we present some initial experimental results.

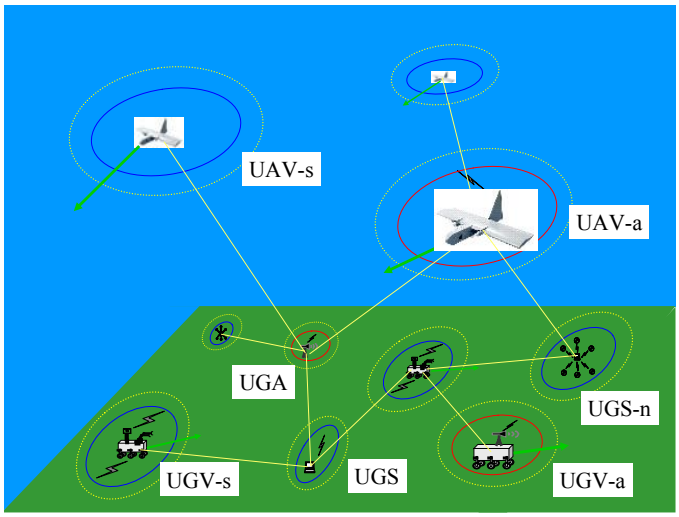


Figure 1: Dynamic surveillance network.

2. THE DSN CONCEPT

The DSN concept proposes to deconflict the problem of data retrieval from remote sensors and the simultaneous requirement for personnel stand-off. Increasingly, unattended, low-power, long-life sensor packages are being deployed in dangerous (e.g., battlefield) environments. Because of power requirements and the need for stealth, these sensor packages are unable to transmit their sensor data long distances (e.g., to satellites or stand-off aircraft). However, the risk is often too high to use personnel or manned vehicles to gain proximity to these sensors. With the Dynamic Surveillance Network, stand-in unmanned aerial vehicles (UAVs) are used to exfiltrate the sensor data from these sensor packages.

In the specific DSN scenario considered here, two UAVs execute a cooperative flight pattern to find previously-deployed unattended ground sensors[2], collect data from these sensors, and then relay the sensor data to a ground station. Initially, the UAVs cooperate to fly a raster scan search throughout the target area. A sensor package is "found" when the UAV establishes a communication link with that sensor package. Once the sensors are located, the UAVs cooperate to establish flight patterns that allow the UAVs to collect sensor data as efficiently as possible. As conditions change (e.g., a sensor or UAV fails), the UAVs dynamically reconfigure their flight patterns. A typical mission scenario is illustrated in Fig 2. Figure 2a shows a distribution of sensor clusters throughout an urban area. Each cluster consists of a number of sensors tied to a single communications node that is used to offload data from the cluster. Typically, these clusters are isolated, and cannot communicate with each other due to environmental constraints and power requirements. Figure 2b shows two UAVs flying a raster scan above the target area where the sensor clusters have been distributed. After covering their respective areas, the two vehicles share their information, and map the locations of all the sensor clusters. In Fig. 2c, the UAVs cooperate to reconfigure their positions and flight patterns in order to optimize data exfiltration from the sensor clusters. When conditions change, the UAVs will adapt and reconfigure. One of these condition changes is shown in Fig 2d. One UAV returns to base to deliver the exfiltrated sensor data to the UAV ground station, and the remaining UAV reconfigures its position and flight pattern to optimize data collection.

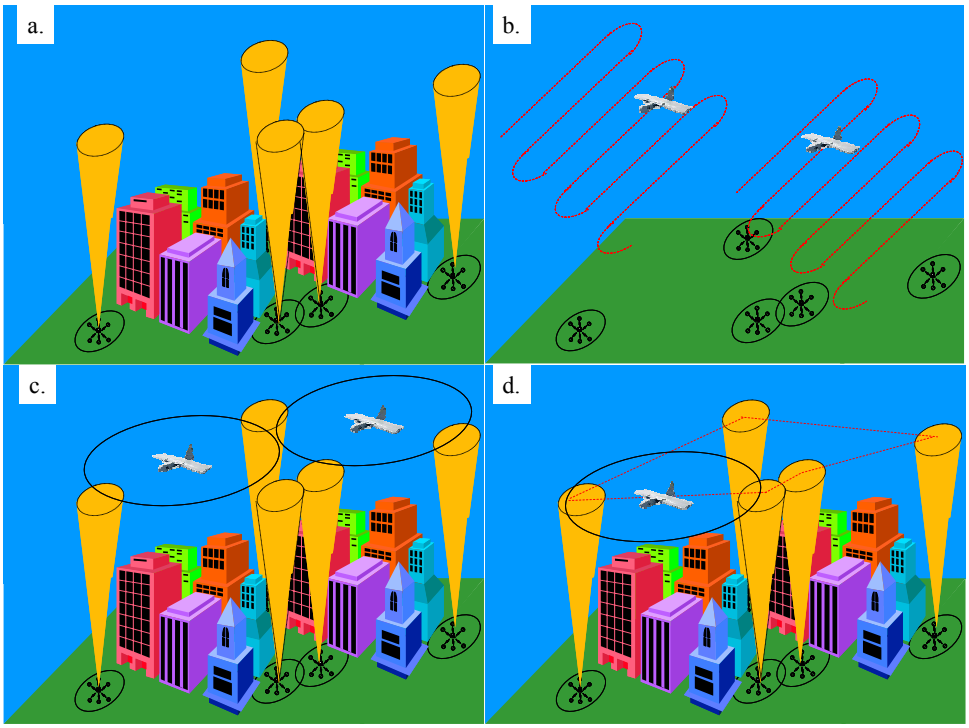


Figure 2: Graphical depiction of DSN concept.

3. DSN COOPERATIVE ALGORITHMS

In the DSN concept the UAVs automatically reconfigure as needed during this mission using a dynamic system strategy known as consensus negotiation. This approach to distributed adaptation and reasoning is based on sophisticated yet scalable mathematical algorithms that are motivated by the idea of a consensus variable [1]. To coordinate complex behaviors, some information must be shared by agents in the network. The minimal amount of information required is

[2] In our testing we are simply using 802.11b-enabled laptops as prototypical sensors. However, we have developed a system concept for a low-power, easily-deployed vehicle detection capability along a roadside in a cluttered urban environment based on clusters of networked magnetometer/accelerometer sensors for tripwire detection. Individual sensors in a cluster communicate to a dedicated base/bridge node using a TDMA protocol in a star topology. Each cluster's dedicated base/bridge node communicates to the "outside world" via an IEEE 802.11b wireless link.

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 the other agents with whom the agent is able to communicate. Through proper definition of the consensus variable, and specification of rules for updating the value of this variable, it is possible for the consensus variable to converge between the communicating agents. This consensus variable approach is applied to the problem of adaptive scheduling of multiple agents, in this case, the two UAVs.

Recently we have developed a unique approach to optimization in wireless sensor networks using a *consensus protocol* paradigm [2]. Specifically, suppose we have *N* nodes in a wireless sensor network, as shown in Fig. 3. Figure 3 can be thought of as depicting the physical communication topology for a wireless sensor network such as that of Fig. 1. We may think of this topology as a partially-connected graph (either directed or bidirectional), where the direction of an edge represents the direction of the information transfer between two connected nodes. We further suppose that each entity has a state that evolves with time and reflects something of importance to the global functionality of the system. For instance, the state of a node could represent its position in space, its belief about the state of other entities, the value of a decision variable, or all of the above.

Next, suppose ξ represents some global consensus variable of interest. Let each node have a local value of the variable, given as ξ_i. Let N_i be the set of all nodes with whom node *i* can directly communicate (i.e., its nearest neighbors - note that the graph as depicted in Fig. 3 is not fully-connected in an "all-to-all" sense) and each node updates its local value based on the values of the nodes with whom they communicate, using the rule:

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

where K_{ij} is a connection gain. It has been shown that as long as there is a path from at least one node to every other node, then all the local values ξ_i will converge to a common value [1]. The key point is that this consensus variable formulation does not require global communications. Its power lies in the fact that sharing only the consensus variable from "neighbor-to-neighbor" is enough under assumptions of the existence of a spanning tree in the communication topology.

The consensus variable idea is primarily concerned with information sharing. The next question to ask is how this can be exploited for distributed signal processing. Assume that the distributed signal processing problem can be can be expressed as an optimization problem with a global performance function *f*, such as $x^* = \min f(x)$ for some vector *x*. If we place a local estimate of *x* at each node of a sensor network and employ a consensus variable-like algorithm of the form

$$x_i = \min f(\varsigma_i)$$
$$\dot{\xi}_i = -\sum_{N_i} K_{ij}(\xi_i - \xi_j) + (x_i - \varsigma_i)$$

then under some appropriate assumptions it can be shown that $\xi_i \to x_i \to x^*$. That is, it is possible to compute some aspect of a problem at 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, with the result that the distributed system will converge to the global solution.

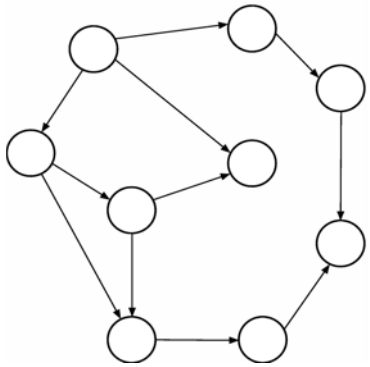


Figure 3: Wireless sensor network physical communication topology.

In this paper space precludes providing the details. However, we point out briefly that we have developed a formulation of the function $f(\varsigma_i)$ that solves the problem of assigning *N* resources to *M* targets. This algorithm will be described in more detail in [3] and is implemented in a general functional architecture that was introduced in [4]. The interested reader can contact us for these references.

4. EXPERIMENTAL HARDWARE

4.1 Vehicle

The vehicle used for our experimental tests are called TAMs, short for Trans-Atlantic Model [5]. The TAMs, have a 1.8 m wingspan and 1.8 m fuselage length, weigh 5 kg with full payload suite and fuel, and can fly approximately 1.5 hrs. with its current fuel capacity. The engine is a modified O.S. Engines FS-61, 10-cc, four-cycle engine that produces about 0.2 hp. A white gas/industrial lubricant mix is used as fuel. The average maximum speed in still air at standard temperature and pressure is 22 m/s. Typical operational altitudes are 150 to 1000 m. The TAMs are sailplanes, and thus are optimized for long duration, but not for high performance. A TAM is shown in Fig. 4.

Figure 4: TAM-11 in flight

4.2 UAV Payload

Onboard Piccolo autopilots from Cloud Cap Technologies enable autonomous flight of the air vehicles. These autopilots typically operate using waypoint navigation. Flight plans can be entered a priori, and can also be dynamically changed during flight. The vehicles are hand-launched, and landed manually, in remote control (RC) mode (i.e., there is no autonomous take-off or landing capability). A standard Futaba RC console is used for manual control.

The Piccolo weighs 212 g, measures 122 mm × 61 mm × 38 mm, and consumes 3.6 W peak power (a photo of the Piccolo is shown in Figure 5). It features an integrated GPS unit, three-axis rate and acceleration sensors, dynamic and static pressure sensors, 10 servo control outputs, and a serial port for external interface. The autopilot communicates with a Piccolo ground station over a two-way 900 MHz link. Typically, the downlink is used during our tests only to send inertial and geolocation data from the air vehicles to the Piccolo ground station. That is, this data link is generally used only for test diagnostics, and is not part of our operational scenario. The uplink, which is used to send manual control signals to the vehicle through the autopilot, is used only when problems arise with the RC controller.

In manual mode, a Futaba RC console sends pulse width modulated (PWM) control signals over a 72 MHz FM link to an onboard Futaba receiver. Typically, manual mode is used only during take-off, landing, and vehicle checkout operations. A dedicated RC channel (CH. 5) is used to toggle between manual and autonomous mode via an onboard custom-designed switch circuit. This configuration allows the pilot to quickly regain manual control at any time during the autonomous flight operations. To enable this safety option, the Futaba RC console must maintain radio link with the vehicle throughout the test area (note, however, that manual control is disabled while the vehicle is in autonomous mode).

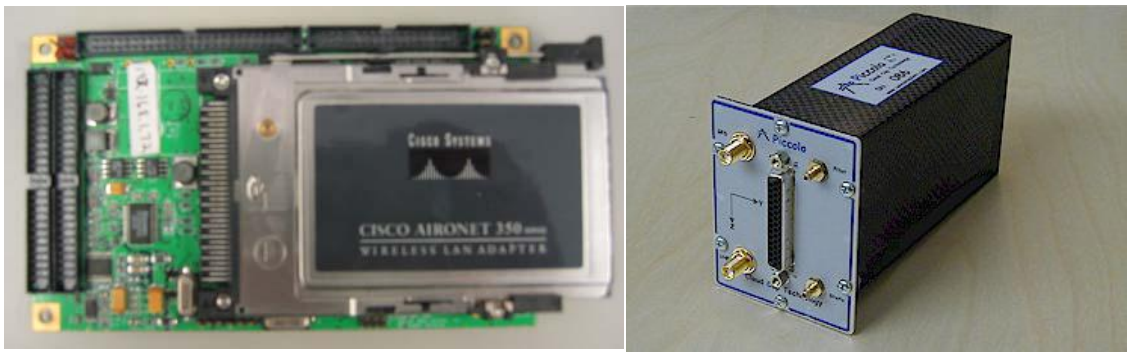


Figure 5: ADS BitsyPlus board with Cisco client card plug-in (left), and Piccolo autopilot (right).

A commercial-off-the-shelf (COTS) based wireless network communications payload has been developed for the UAVs that includes an Applied Data Systems (ADS) BitsyPlus single board computer and a Cisco AIR-LMC352 Aironet 350 client card adapter (see Fig. 5). The single board computer features a 206 MHz StrongARM reduced instruction set computer (RISC) processor, 32 MB Flash memory, and 64 MB RAM. The operating system is Windows CE. Together, the supervisor board and WLAN card have a footprint of 76 mm × 127 mm, weigh 100 g, and consume 4.25 W peak power (2 W attributed to the supervisor, 2.25 W to the WLAN card in transmit mode).

The single board computer processor and onboard RAM support integration of finite state automata (FSA) that are used to implement the consensus negotiation strategy and the communication framework. The computer directs vehicle behavior by sending navigation commands (GPS waypoint updates) over a serial interface to the Piccolo autopilot in flight. The computer also contains the drivers for the Cisco network card. The Cisco client cards are IEEE 802.11b WLAN devices. For the tests described below the network devices were used to form an independent basic service set (IBSS), better known as an ad hoc network. The Cisco client cards transmit at signal levels scalable from 1mW (0dBm) to 100mW (20dBm). Receive sensitivity depends on the data rate: -85 dBm for 11 Mbps, -89 dBm for 5.5 Mbps, -91 dBm for 2 Mbps, and -94 dBm for 1 Mbps. These client card adapters have two antenna ports for external diversity antennas, or for a single external antenna. The card interfaces to the main processor board through a PCMCIA interface bus. These Cisco client cards are used as the network devices for all nodes on the network, both airborne and ground-based.

4.3 System Architecture

The electronics and communications architecture for our vehicles is shown in Fig. 6. As shown in the figure, there are four wireless links used during these tests. However, the 900 MHz Piccolo link is typically used only for vehicle diagnostics during testing and demonstration. That is, this link is not essential to the DSN concept implementation, and is planned to be eliminated in future phases of development. Furthermore, the 72 MHz RC link is used only for take-off, landing, and vehicle checkout operations. If an autonomous take-off and landing capability is established, this link also can be eliminated.

The Piccolo autopilot ground station (AGS) is comprised of the Piccolo ground station receiver, a laptop computer, an omnidirectional UHF antenna, and an RC controller. The autopilot ground station configuration is shown in Fig. 7. This ground station is used to monitor position and flight dynamics during flight, and can be used to manually fly the vehicle if problems arise with the 72 MHz RC controller. The ground station can also be used to send waypoints to the autopilot, though for these tests the intent was to send these waypoints from the onboard computer over a serial cable.

The sensor clusters were simulated using three test sensor nodes, each comprised of a laptop, Cisco WLAN card, and a 2.4GHz directional antenna. This configuration is shown in Fig. 8. For these tests, the antenna mainbeam was pointed at 0º (skyward). The 2.4GHz directional antenna is a parabolic grid antenna with a gain of 24dBi and a 3dB beam-width of 7.5º. The purpose of this antenna is to simulate the link condition in which none of the test sensor nodes are in direct communications with each other, and the test sensor node signals are above the UAV receiver sensitivity thresholds only in sub-zones within the test area (i.e., not throughout the entire test area). The radiation pattern and performance characteristics of the antenna are shown in Fig. 9.

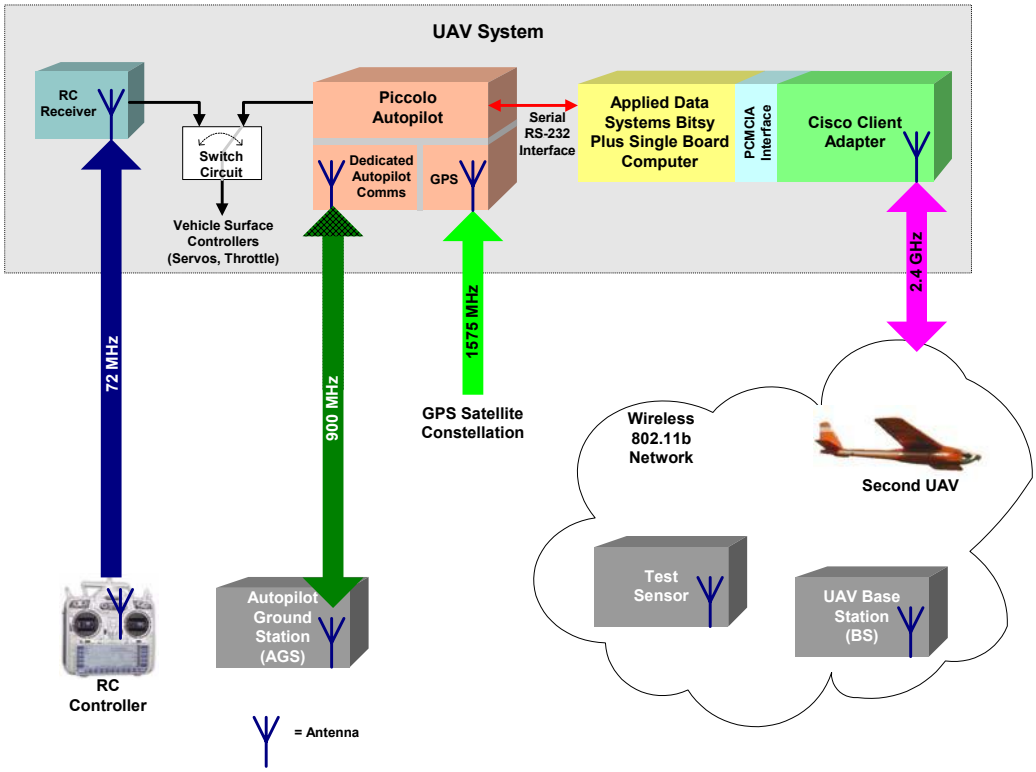


Figure 6: Vehicle hardware architecture and system communications architecture for these tests.

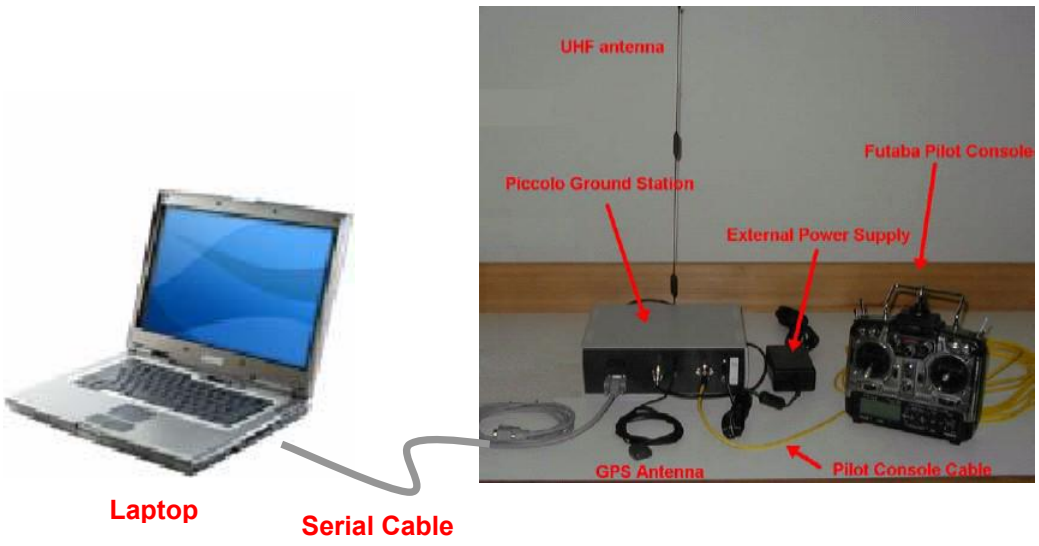


Figure 7: Autopilot ground station

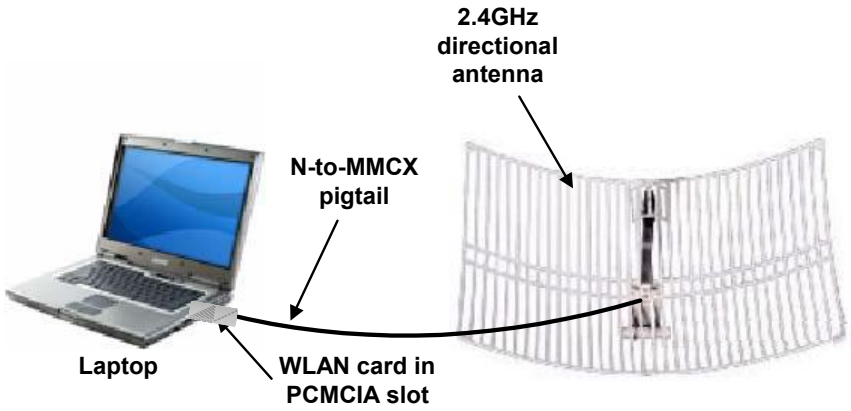
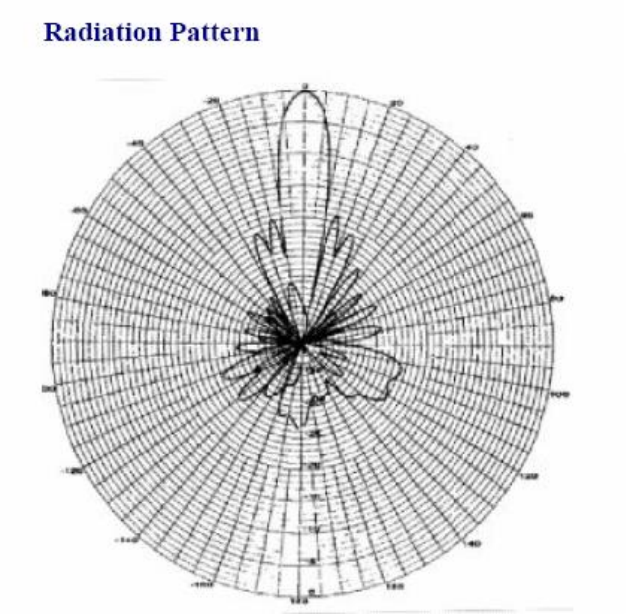


Figure 8: Test sensor node configuration.



Antenna Performance	
Frequency	2.4 - 2.485 GHz
Gain	24 dBi
VSWR	1.4:1 @ 2.45GHz
Polarization	Vertical or Horizontal
Impedance	50 Ohms
Elevation Adjustment	60° in 10 increments
-3 dB Beamwidth	7.5°
Front to Back Ratio	31 dB
Cross Polarity Rejection	26 dB
Physical Characteristics	
Dimensions	24 in x 39 in x 15 in (610mm x 915mm x 381mm)
Weight	5.4 lbs (2.43 kg)
Wind Load	141.5 lbs @ 120 mph
RF Connector	N type female (available with custom connectors)
Material	Cast magnesium alloy
Mounting	Stainless steel 1 - 2 in (25.4 - 50.8 mm)

Figure 9: Directional antenna characteristics

Preliminary network link estimations were conducted to determine the signal coverage of each of the test sensor nodes. These calculations showed that a TAM UAV flying at an altitude of 150 m can establish a network link only if it is within approximately 70 m lateral distance of the test sensor node. The full high-speed (11Mbps) link is available to the UAV within approximately 34 m lateral distance of the sensor node cluster. Some assumptions made for these calculations include a 0 dBi aircraft antenna gain throughout flight, 1 mW (0 dBm) ground WLAN transmit power, 100 mW (20 dBm) airborne WLAN transmit power, and 5 dBm insertion/cable losses total (for both ground and airborne equipment). Also, because the tests were conducted with unobstructed line-of-sight between the air vehicles and ground test sensor nodes, ground topography, buildings, and obstructions were not considered. Furthermore, Fresnel zones, multipath, interference, and noise were not considered.

5. PRELIMINARY TEST RESULTS

To date we have not completed a cooperative test flight, but have successfully shown the ability to fly our hardware autonomously while communicating with the laptop-based sensor emulation stations. Complete tests with two cooperative UAVs will be described in [3]. Here we describe the test results obtained to date.

5.1 Flight Pattern and Sensor Node Locations

Fig. 10 shows the primary DSN test flight area and approximate planned flight path. Three test sensor node locations, denoted S1, S2, and S3 in the figure, were chosen to ensure that this search pattern would establish network connectivity with each node at least some point along the flight path. Also shown on the figure are the received signal zones that had been derived previously: the green rings enclose the zone of minimal (1 Mbps) network connectivity, and the purple rings enclose the zone of high data rate (11 Mbps) connectivity. Figure 10 also shows the location of the autopilot ground station (AGS). The autopilot ground station is shown in blue to indicate that it is not part of the network, but rather operates on a dedicated 900 MHz link.

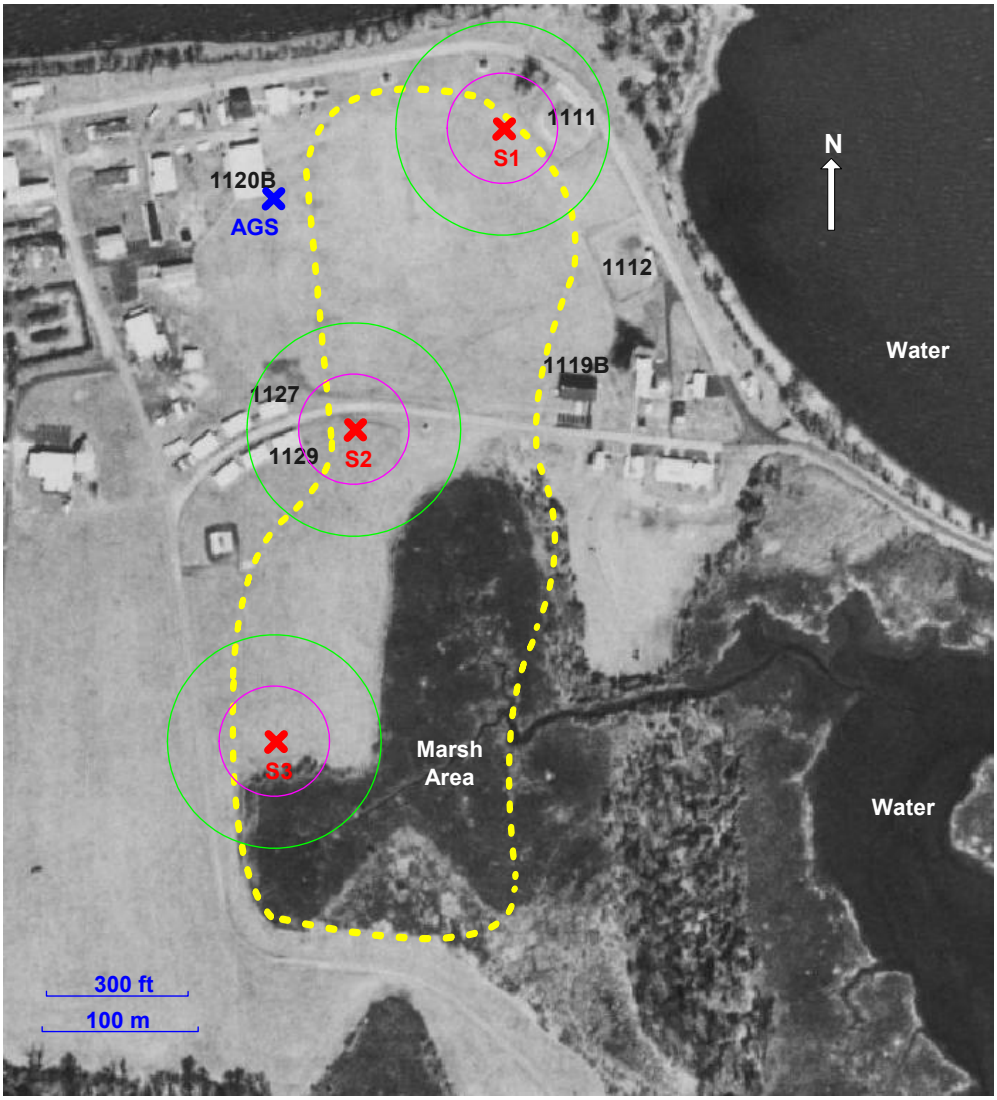


Figure 10: Test site with test sensor node locations.

5.2 Test Results

The two primary objectives of our preliminary tests were to confirm the ability of the onboard computer to send waypoints to the autopilot over the serial interface and to map network connectivity throughout the test field. To test the first objective, a mission was programmed by the onboard computer over the serial interface. The test of this objective was intended to validate the computer-to-autopilot interface, as well as the JHU/APL algorithms used to program the autopilot. Programming occurred after boot-up of the computer and autopilot, and was instantiated completely independently of the Piccolo ground station. Figure 11 shows the trajectory resulting from one such test using just two waypoints. In this figure waypoints are shown as red squares, the laptops that are emulating sensor clusters are colored triangles. The UAV path is shown in blue.

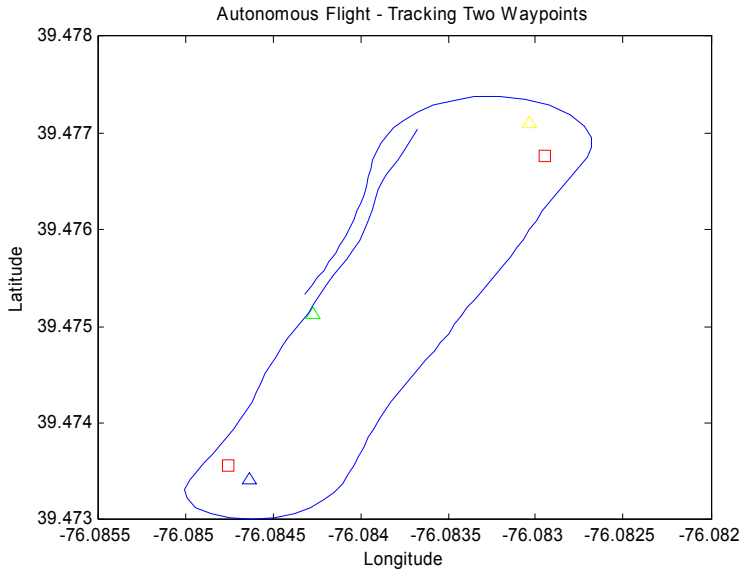


Figure 11: Autonomous waypoint tracking.

Figures 12 and 13 show the results of tests to establish the ability to map network connectivity. In these figures colored stars show the locations where connectivity along the path could be established (determined by a successful response when "pinging" the address of each laptop). The colors are coded to match the laptop's colored triangles. Thus a green star means from that location the UAV was able to successfully communicate with the laptop at the location indicated by a green triangle. Figure 12 shows a map generated by approximately ten minutes of flight. To give a better idea of the notion of a connectivity map, the flight path is not shown. In Figure 13 we also show the flight path for a different test run, along with the associated "pings." In this test only two laptop/sensors were active. It should be noted that the resolution of our tests were determined by the fact that our code worked according to the following sequence:

1) Read GPS
2) Ping Laptop #1 with a 1 sec time-out
3) Write result
4) Ping Laptop #2 with a 1 sec time-out
5) Write result
6) Ping Laptop #3 with a 1 sec time-out
7) Write result
8) GOTO 1)

As a result, it was possible to travel at as long as 3 seconds between GPS reading. Because we were often traveling 20 m/sec, our spatial mapping resolution could be as coarse as 60 meters. In future tests we are modifying our code to give a better spatial resolution.

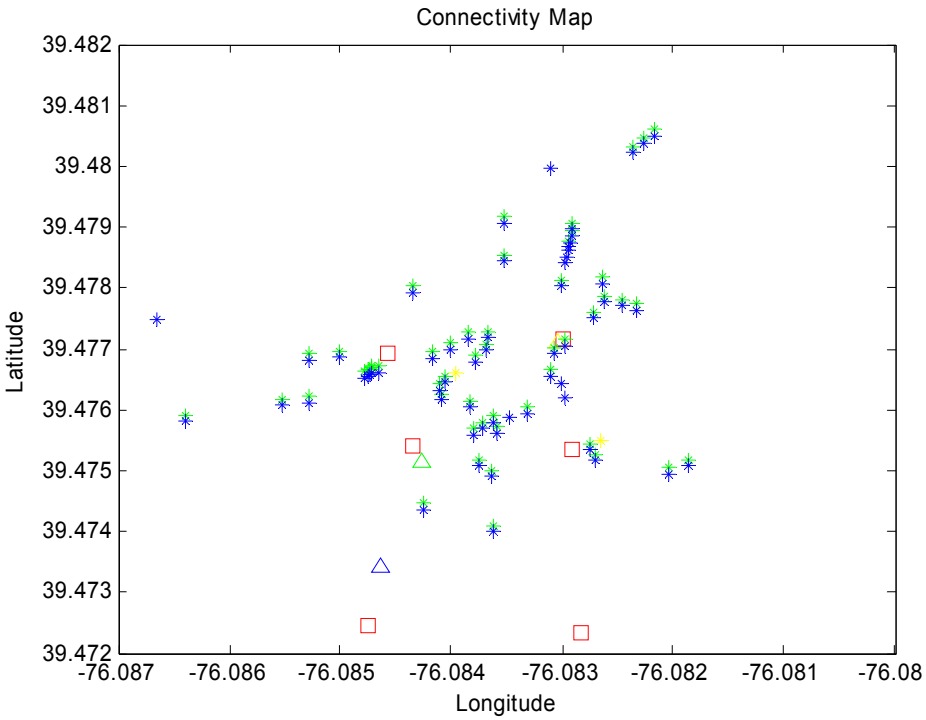


Figure 12: Connectivity map.

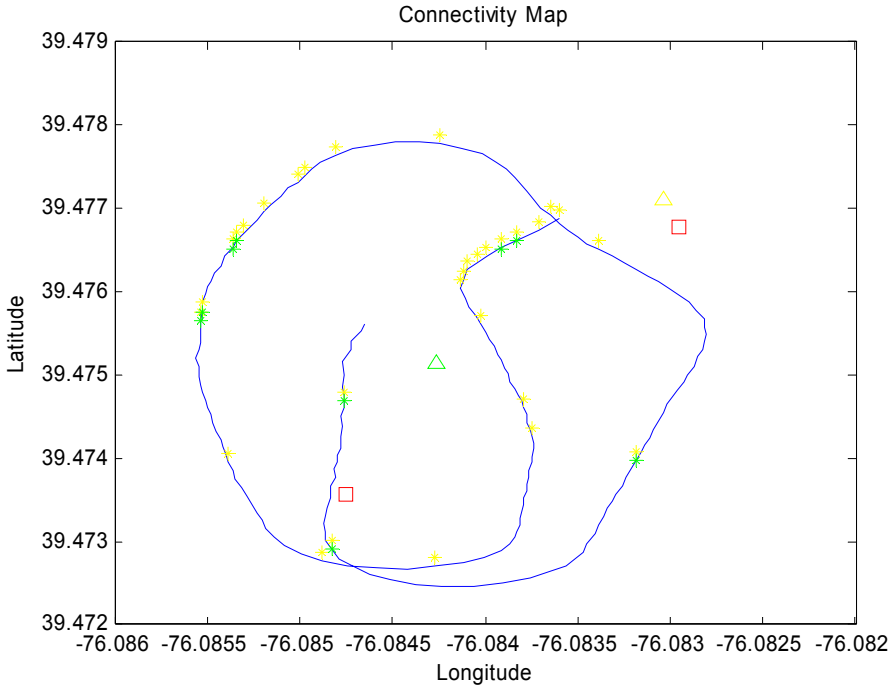


Figure 13: Connectivity map with overlaid flight path (different test than Fig. 12).

6. CONCLUSION

In this paper we have present the idea of a dynamic surveillance network and a specific instance of using such a network whereby multiple, cooperating UAVs exfiltrate data from unattended ground sensors. We described our algorithmic approach to cooperative behavior, presented our hardware and gave some preliminary results. Using a TAM UAV, outfitted with Piccolo autopilots and a flight autonomy board, we showed the ability to achieve autonomous waypoint navigation and to build network connectivity maps. Future efforts will focus on completing a two-UAV demonstration of the data exfiltration scenario.

REFERENCES

[1] "Coordination Variables and Consensus Building in Multiple Vehicle Systems," W. Ren, Randall W. Beard, and Timothy W. McLain, in *Proceedings of the Block Island Workshop on Cooperative Control*, Springer-Verlag Series: Lecture Notes in Control and Information Sciences, 2003.

[2] "Forced and constrained consensus among cooperating agents," Kevin L. Moore, and Dennis Lucarelli, in *2005 IEEE International Conference Conference on Networking, Sensing, and Control*, Tuscon, AZ, March 2005.

[3] "Cooperative UAVs for Remote Data Collection and Relay," Kevin L. Moore, Michael R. White, Robert Bamberger, and David P. Watson, accepted to appear in *Proceedings of AUVSI Unmanned Systems North America 2005*, Baltimore, MD, June 2005.

[4] "Consensus variable approach to decentralized adaptive scheduling," Kevin L. Moore, and Dennis Lucarelli, in *5th International Conference on Cooperative Control and Optimization*, Gainsville, FL, January 2005.

[5] "Wireless Network Communications Architecture for Swarms of Small UAVs", R. Bamberger, et. al., *Proceedings of the AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, AIAA-2004-6594, Chicago, IL, Sep. 20-23, 2004.