

# Hierarchical Distributed Control for Search and Tracking by Heterogeneous Aerial Robot Networks

Jack Elston

Aerospace Engineering Sciences Department  
University of Colorado  
Boulder, Colorado 80309

Eric W. Frew

Aerospace Engineering Sciences Department  
University of Colorado  
Boulder, Colorado 80309

**Abstract**—This paper presents a hierarchical control architecture that enables cooperative surveillance by a heterogeneous aerial robot network comprised of mothership unmanned aircraft and daughtership micro air vehicles. Combining the endurance, range, and processing capabilities of the motherships with the stealth, flexibility, and maneuverability of swarms of daughterships enables robust control of aerial robot networks conducting collaborative operations. The hierarchical control structure decomposes the system into components that take advantage of the abilities of the different types of vehicles. The motherships act as distributed databases, fusion centers, negotiation agents, and task supervisors while daughtership control is achieved using cooperative vector field tracking. This paper describes the overall architecture and then focuses on the assignment and tracking algorithms used once sub-teams of daughtership vehicles have been deployed. A summary of the communication, command, and control structure of a heterogeneous unmanned aircraft system is also given in this paper along with hardware-in-the-loop and software simulation results verifying several components of the distributed control architecture.

## I. INTRODUCTION

Heterogeneous aerial robotic networks that combine the long endurance, range, and processing capabilities of mothership unmanned aircraft with the stealth, flexibility, and maneuverability of swarms of daughtership micro air vehicles (MAVs) are ideal for cooperative search, acquisition, and tracking (CSAT) in urban environments and rough terrain [1] [2]. Micro air vehicles offer greater flexibility, maneuverability, and stealth compared to larger unmanned aircraft (UA) systems. However, a fundamental drawback of MAV systems is their short range, low endurance, and small payload capacity. Current MAV systems typically have limited operation of 1-2 hours maximum flight time. While this amount of flight time is adequate for "over the hill" surveillance, it limits the use of swarms to small-scale engagements. Deployment from larger mothership vehicles enables the insertion of MAV swarms into environments of extended size and is the ideal strategy for CSAT applications.

In addition to extending the footprint of operation of MAV teams, the mothership/daughtership (MS/DS) concept increases system performance by combining the complementary capabilities of the MS and DS vehicles. The payload capabilities of MAV platforms fundamentally limit the amount of sensor information, communication, and processing that

can be achieved by these systems. In contrast, the size of the mothership vehicles allows them to carry systems that can communicate over larger distances and process more information at higher rates than the DS vehicles. Thus, coordinated control of micro air vehicle teams deployed in MS/DS architectures exploits the communication and processing capabilities of the mothership vehicles to minimize the amount of communication, estimation, planning, and coordination performed by the DS vehicles.

This paper describes a hierarchical control architecture that enables cooperative search, acquisition, and tracking by a heterogeneous aerial robot network operating in a mothership/daughtership configuration. A high-level description of the control architecture is provided that highlights complementary aspects of the mothership and daughtership vehicles. Details are provided for assignment and cooperative tracking algorithms utilized by daughtership sub-teams once they have been deployed. Additionally, this paper describes the communication, command, and control structure of a heterogeneous aerial robot network developed to validate the MS/DS concept of operations. Software simulations are provided to demonstrate aspects of the control structure while hardware-in-the-loop simulations highlight components of the experimental platform.

## II. HIERARCHICAL COOPERATIVE SEARCH, ACQUISITION AND TRACKING ARCHITECTURE

Consider a heterogeneous unmanned aircraft system denoted by the set  $H$  consisting of two main types of aircraft comprising the (disjoint) sets  $M \subset H$  of *mothership* vehicles and  $D \subset H$  of *daughtership* vehicles. Let  $m_i \in M, i = [1, \dots, n_m]$  denote the  $i_{th}$  mothership vehicle in the set  $M$ . Each mothership is assigned a subset  $D_i \subseteq D$  of daughtership vehicles to which it communicates where  $d_{i,j} \in D_i, j = [1, \dots, n_i]$  is the  $j_{th}$  vehicle in  $D_i$ . We assume that each daughtership communicates with one and only one mothership vehicle and each mothership  $m_i$  acts as the gateway to the subnetwork defined by the vehicles in  $D_i$ . Communication between the mothership  $m_i$  and daughterships  $d_{i,j}$  occurs directly or over a multi-hop connection.

This work assumes UA are equipped with a low-level flight control system that provides roll, pitch, and yaw stability of the aircraft as well as velocity tracking and altitude-hold

functions. For aircraft guidance, the flight control system accepts speed, climb rate, and turn rate commands. Using this command structure, the model presented to the guidance layer of the  $i^{th}$  UA is a kinematic model:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\psi}_i \\ \dot{h}_i \end{bmatrix} = \begin{bmatrix} u_{1,i} \cos(\psi_i) + W_x \\ u_{1,i} \sin(\psi_i) + W_y \\ u_{2,i} \\ u_{3,i} \end{bmatrix} \quad \begin{aligned} v_{min} &\leq u_{1,i} \leq v_{max} \\ |u_{2,i}| &\leq \omega_{max} \\ |u_{3,i}| &\leq h_{max} \end{aligned} \quad (1)$$

where  $\mathbf{x}_i = [x, y, h] \in \mathbb{R}^3$  is the inertial position of the aircraft,  $\psi \in [0, 2\pi)$  is the aircraft yaw angle,  $[W_x, W_y] \in \mathbb{R}^2$  are the components of the steady background wind velocity, and  $[u_1, u_2, u_3]$  are respectively the commanded air speed (m/sec), turning rate (rad/sec), and climb rate (m/sec), which are constrained to the given limits.

Typical sensors on unmanned aircraft include electro-optical cameras, ground target motion indicators, radar, and signal strength analyzers. For this work we consider the field of view of the sensor and relative position measurements that can be taken with them. We define the function  $F_s(\mathbf{x}_k)$  as the sensor footprint, i.e. area of the environment from which a sensor can receive a (position or velocity) measurement  $\mathbf{z}_{i,j}$  of some target  $t_j$ . Given a target located at position  $\mathbf{p}_j$  the measurement obtained by a UA at position  $\mathbf{p}_i$  includes the range to the target  $r_{ij}$  and the target bearing  $\beta_{ij}$  with zero-mean Gaussian noise terms with covariances  $\sigma_r$  and  $\sigma_{beta}$ , respectively. In general, the sensor coverage of a mothership vehicles is greater than that of the daughterships, i.e.  $F_{DS}(\mathbf{x}_k) \subset F_{MS}(\mathbf{x}_k)$  while the accuracy of the relative positions sensors on the DS is better since they are closer to the targets, i.e.  $\sigma_{DS} < \sigma_{MS}$ .

Figure 1 depicts a schematic representation of the hierarchical coordination architecture. At the highest level, the coordination layers of individual mothership vehicles deliberate over the AUGNet meshed network in order to allocate specific tasks through negotiation. Once tasks have been allocated to each mothership, daughtership vehicles are directly commanded through an existing hybrid control policy. Through this policy, tasks are allocated from the Coordination Layer on the mothership vehicle to the Trajectory Guidance Layer on each daughtership vehicle. Here, each task is converted into a trajectory which is passed to the lowest Flight Control layer which commands the vehicles aerodynamic surfaces.

Target assignment for the DS vehicles is integrated with area search and broken down into a several steps. This decomposition is motivated by the need to keep the complex distributed resource allocation algorithms off the computationally-limited DS vehicles. The main layers of the distributed target assignment are: 1.) negotiation between MS vehicles to "claim" responsibility for ground targets; 2.) determination of which targets to track; 3.) assignment of active targets to available DS; and 4.) area search by the remaining DS. Ownership of tasks by the motherships will be determined through deliberative cooperation between

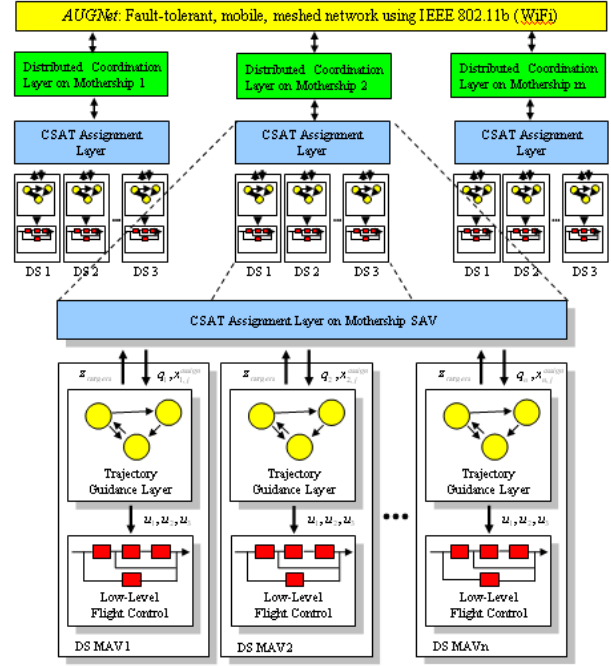


Fig. 1. Hierarchical cooperative search, acquisition, and tracking architecture.

vehicles using a prioritized, iterative resource allocation scheme. However, we assume here that tasks have already been claimed by the MS robots.

DS assignment to tracking and search tasks is achieved using a new integrated algorithm [3]. In this work the goal of the tracking system is to maintain target positioning error within bounds so targets must be repeatedly visited. In order to insure that targets are revisited before they violate given uncertainty bounds, a new metric is developed that quantifies a vehicles ability to reach a target in time

$$e_2(\mathbf{x}_{i,k}, \bar{\mathbf{p}}_t, \mathbf{P}_k) = \frac{v_{i,ave}}{\dot{\rho}_{ave}(\mathbf{P}_k)} \cdot (\rho^* - \rho_k) - d_{i,j}. \quad (2)$$

where  $d_{i,j}$  is the distance between the DS and the target,  $v_{i,ave}$  is the average speed of the robot during target intercept,  $\rho_k$  is the square root of the maximum singular value of the position component of the target estimate covariance matrix,  $\rho^*$  is the specified bound on  $\rho$ , and  $\dot{\rho}_{ave}(\mathbf{P}_k)$  is the average time rate of change of  $\rho$ . Interception of the  $j^{th}$  target by the  $i^{th}$  robot is then possible as long as  $e_2 \geq 0$ . This new metric is used to activate targets for tracking and as the cost metric in the assignment algorithm that uses binary linear programming.

DS vehicles not performing target tracking are assigned to search the environment. Search is achieved using task assignment based on a decomposed coverage map. The cells with highest coverage variable  $a_c$ , which represents the probability that a target is in the cell, are assigned to the remaining DS using binary linear programming. Details of the integrated assignment process can be found in Ref [3].

The daughtership guidance layer commands inputs for the

low-level flight control system are based on a Lyapunov vector field approach developed specifically for micro air vehicles [4], [5]. The basic motion primitive is a vector field that directs the DS to orbit around a potentially moving ground target at position  $\mathbf{p}_t$  [4], [5]. The vector field provides a reference velocity vector which is tracked by feedforward and proportional feedback from the heading angle error to command the vehicle turning rate. The stable behavior of the vector field approach enables robust abstraction of assignment tasks to higher levels in the hierarchical CSAT scheme.

Consider the Lyapunov function  $V(r) = (r^2 - r_d^2)^2$ , where  $r$  is the radial distance of the DS from the target position. The total time derivative of  $V$  can be specified to be non-positive by choosing desired relative vehicle velocity  $\dot{\mathbf{p}} = \dot{\mathbf{p}}_d = g(\mathbf{p})$  according to the guidance vector field

$$g(\mathbf{p}) = \frac{-\alpha u_1}{r(r^2 + r_d^2)} \begin{bmatrix} x_{ij} \cdot (r^2 - r_d^2) + y_{ij} \cdot (2r \cdot r_d) \\ y_{ij} \cdot (r^2 - r_d^2) - x_{ij} \cdot (2r \cdot r_d) \end{bmatrix} \quad (3)$$

where  $\alpha$  is non-negative [4]. When  $r \gg r_d$ , the vector field points to the loiter circle center and the field veers away when approaching the circle to smoothly entrain the motion into a left turn loiter.

Phase coordination of multiple robots around the desired contour is produced by a second control law which adjusts the speed of the vehicles (within limits). For two cooperating robots with phase angles  $\theta_1$  and  $\theta_2$  defined relative to the instantaneous target location, the speed commands

$$\begin{aligned} u_{1,1} &= k_1(\theta_2 - \theta_1 - \theta_D) + v_0 \\ u_{1,2} &= -k_1(\theta_2 - \theta_1 - \theta_D) + v_0 \end{aligned} \quad (4)$$

drive the relative angle  $\theta_2 - \theta_1$  to the desired phase offset  $\theta_D$  [4]. An important characteristic of this algorithm is that only a single additional variable,  $\theta_i$ , needs to be communicated between vehicles in order to enable coordination.

When multiple DS are tracking an uncertain moving target, the covariance matrix  $\mathbf{P}$  of the estimate position error can be used to define an elliptical orbit pattern about the target. In order to insure that the desired stand-off radius  $r_d$  is maintained we first take the singular value decomposition of the matrix

$$\mathbf{S} = n_\sigma \sqrt{\mathbf{P}} = \mathbf{U} \cdot \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \cdot \mathbf{V} \quad (5)$$

and then create the new matrix

$$\mathbf{M} = \mathbf{U} \cdot \begin{bmatrix} \sigma_1 + r_{des} & 0 \\ 0 & \sigma_2 + r_{des} \end{bmatrix} \cdot \mathbf{V} \quad (6)$$

To loiter about the new elliptical pattern we define the new vector field  $g_P(\mathbf{p}) = \mathbf{M} \cdot g(\mathbf{M}^{-1} \cdot \mathbf{p})$ .

The trajectory guidance layer is currently made up of the several modes derived from the vector field algorithms. The LOITER, FOLLOW, and PROTECT modes are cooperative tracking controllers for stationary targets (waypoints), moving targets, and friendly targets, respectively. The SEARCH mode represents the default action of the DS vehicle based on the last commanded location from the assignment layer.

Since the goal of the system is to track each target to within the prescribed bounds, a mode switch occurs once  $\rho_t \leq \rho_{min}^*$ .

### III. HETEROGENEOUS AERIAL ROBOT NETWORK ARCHITECTURE

The implementation of the robot network, referred to as the Heterogeneous Unmanned Aircraft System (HUAS), consists of multiple small Ares UA and miniature CU MAVs connected through a multi-tier airborne network. The system is designed so that each UA can operate autonomously and recognize the capabilities of other friendlies in the area in order to self-organize cooperative groups. Should input to the system from a remote operator be desired, stations located on site (within communication range of the meshed network) or off-site can at any time be added to the network. Command and control data from outside the deployed mobile ad-hoc network is sent to a network gateway which provides address translation and transmits the data to the vehicles. Using peer-to-peer routing protocols data hops through the network to the intended source, taking multiple hops between heterogeneous nodes as needed. Telemetry and payload sensor data are provided in a publish-subscribe manner, and can be subscribed to by any network participant.

The architecture of the HUAS was designed using a bottom-up layered approach (Figure 2). This approach allows for the design of UA and supporting systems to reflect and enhance design decisions made at the network layer. Given that the success of the HUAS system is based heavily on networked communications, this approach guarantees that the requirements of the high level cooperative algorithms can be met by the underlying architecture. Design was conducted by constructing and fully characterizing lower levels before making subsequent design decisions at higher layers. The design layers are: the physical and transport layer, the network protocol and data routing layer, the sensor, communication, and control fusion layer, the application layer networking, and the cooperative control layer.

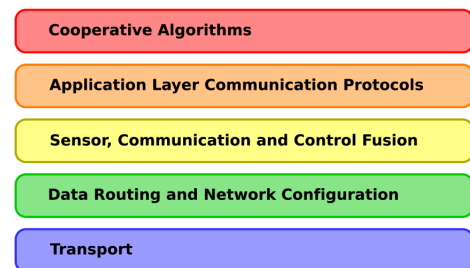


Fig. 2. Effective Layers used for the CU HUAS Design.

#### A. Physical and Transport Layer

Two main aircraft were developed for the HUAS, however the network has been extended to allow for participation by most of the UA in the CU fleet (Figure 3). The first vehicle is the CU Micro Air Vehicle (MAV), which is based

on a modified commercial foam kit with electric propulsion that has an empty weight of approximately 20 oz. At small vehicle scales, avionics mass is a significant limiting factor of vehicle range and endurance. This was minimized by a flight control system which uses only one rate gyro for attitude control, an extremely compact GPS receiver for positioning, and a MEMS pressure sensor for altitude. Two micro-servos on elevons provide the only control surfaces. Details of the avionics system for the MAVS can be found at [6].

MAV networking is provided by the Maxstream XBEE Pro Zigbee radio using the IEEE 802.15.4 protocol. These devices were chosen for their lower power, small size, relatively long transmission range (verified to 2 mi), serial interface compatibility with avionics processors, and packet interface. The radio provides simple point to point or networked functionality, with newer models implementing AODV routing protocols in hardware. Power consumption of the radio is relatively modest compared with powering the aircraft's electric propulsion system.

The second UA is the Ares aircraft which is constructed entirely at CU Boulder. The airframe of the Ares aircraft is based on the layout of the Senior Telemaster, a popular RC model. Custom modifications include an expanded fuselage to accommodate a payload section and conversion from a tail-dragger configuration to one with tricycle gear. The monocoque fuselage, wings, main gear, and horizontal tail are made of a carbon composite laid over carbon-composite laminated plywood bulkheads and ribs. The Ares is powered by a 5-hp, two-stroke engine and has an additional payload capacity of 10 pounds. With a full 1-gal tank, the endurance is estimated to exceed 3 hr. The Piccolo Plus autopilot is used for autonomous flight, with manually controlled takeoff and landing.

The mobile ad-hoc network connecting the Ares UA to one another and dispersed operators is based on the IEEE 802.11b (WiFi) standard, chosen primarily for COTS compatibility. Network software combined with the communication hardware is denoted as the *mesh network radio (MNR)*. The MNR hardware components are a Soekris Engineering Model 4511 single board computer (100-MHz 486 processor, 64-MB RAM, 256-MB flash memory), an Atheros AR5213A chipset mini PCI card, a Fidelity Comtech bidirectional amplifier

with up to 1-W output, and a Garmin Model 35-HVS GPS receiver. The core of each MNR node is identical, only the packaging differs depending on whether the MNR is mounted at a fixed ground site, on a ground vehicle, or in a UA. Flight test results have demonstrated that under typical conditions this system can achieve 1.0 Mbps throughput at a distance up to 4km from the UA to ground, but have also demonstrated drawbacks to using 802.11b in highly dynamic environments [7].

### B. Network Protocol and Data Routing

The network protocol and data routing layer was built on top of the COTS wireless networking standards. Two distinct network tiers, a high-level tier supported by the Ares UA based on IEEE 802.11b (WiFi) and a low-level tier supported by the MAVs using IEEE 802.15.4 (Zigbee), were developed and combined into a single multi-tiered network. The high-level tier is based on the Ad-hoc UA Ground Network (AUGNet) [7] and the low-level tier is based on the CU MAV Sensor Flock concept [8].

The AUGNet system is a wireless multi-hop network designed for highly mobile environments. It includes the ability to collect various network performance metrics and log them to remote servers in real time. The main components of the system include: custom made mesh network radios; network monitoring software and a database that can be accessed during flight experiments from the Internet; routing software using the Click modular router to implement dynamic source routing (DSR); and the CU Ares aircraft. TCP and UDP transport protocols are supported, along with a modified version of the UDP protocol which provides reliable message delivery using node-by-node custody transfer[7]. The AUGNet system has been characterized through various experiments investigating the impact of disconnected groups, air to ground communication, and extended air-to-air networking [7].

Meshed networked communication between MAVs and from the MAVs to the Ares UA is achieved using a custom implementation of the IEEE 802.15.4 protocol. This network stack sits atop the PHY and MAC network layers provided by the IEEE 802.15.4 standard. It consists of a number of additional layers that provide high-mobility ad-hoc networking capabilities, including security functions, mesh networking, and multi-modal (i.e. high data rate, low data rate, and sleep mode) operation. In particular, current networking research developments are directed toward comparison of standard ad-hoc approaches, such as AODV, with new MAV-tailored schemes that exploit asymmetry in the MAV-SAV link capabilities such as high power one-hop SAV-MAV and low-power multi-hop MAV-MAV and MAV-SAV links.

### C. Sensor, Communication, and Control Fusion

The next functional layer in the HUAS architecture utilizes the multi-tiered network to create a system that can fuse sensor, communication, and control information into individual intelligent vehicle platforms. The core of this layer are a



Fig. 3. CU Air Vehicle Fleet and Support Trailer. Vehicles from Left to Right, Top to Bottom are: Ares SAV, CU MAV, Velocity SAV, NexSTAR SAV, Support Trailer, and the MLB Bat UAS.

networked communication, command, and control system; onboard component integration through interface nodes; and flight management for individual vehicles [9].

Onboard flight management of the Ares aircraft is achieved through an embedded architecture that integrates custom and COTS components using intelligent interface nodes. Modularity and module independence are achieved by using high level bus protocols between each interface node. The flight management system (FMS) consists of the Communication subsystem, the Autonomous Flight Control Module, the Computational Module, and a collection of sensors in the Sensor Modules subsystem. The FMS is monitored and controlled from Remote Monitor Stations using the Virtual-Cockpit through AUGNet.

#### D. Application Layer Networking

The application layer networking provides the mechanism for service discovery, data stream subscription, and command issuance over the network. This software enables using more than one groundstation; communications with any node without prior knowledge of IP addresses; interaction with the UA from outside of the adhoc network; and communication between both MAVs and SAVs over the multiple, heterogeneous sub-networks.

Application layer networking is implemented entirely in software and uses an object-oriented approach to define components. The highest level object is the Network Appliance. The Network Appliance is composed of one to many Interface objects, each providing a connection between individual inputs and outputs (ad-hoc network, zig-bee network, www interface, slip interface) and the central Data Distribution object. The Data Distribution object intelligently forwards incoming data to one or many of the Interface objects. Each network appliance also includes a Data Logging object, a Capabilities Manager object (which was added for the service discovery protocol), and a list of known clients. The client list keeps track of all devices the Network Appliance has communicated with and stores pertinent information such as last communication time, last known location, text name or description (e.g "Ares 2"), address, capabilities, etc.

Each interface object, when invoked, registers the capabilities it enables with the Capabilities manager object. It is in this manner that the network appliance object is aware of its various capabilities (autonomous flight, gateway to external networks, etc.). Associated with each capability are data streams that other clients might subscribe to, or commands that can be accepted by the particular interface. This information is broadcast over the network at a fixed interval providing for service discovery to occur, and simple status of each network node to be maintained.

To define services, a description language was needed. Most of the existing languages require XML (which would require too much overhead on the smaller platforms), have too extensive of a library, or do not allow for the proper definition of services. In the current implementation, each UA subsystem is given a 4 bit unique id. Available streams

and accepted commands for the subsystem are defined by two 8 bit bit fields. This 20 bit representation of a set of subsystem capabilities provides enough flexibility to define all of the current subsystems and allows for significant future expansion. Furthermore, by using the combination of the 4 bit subsystem id, and the 4 bit location of the specific accepted command or data stream identifier bit as the header for internal packets, many interesting networking capabilities are enabled such as content-based forwarding.

#### E. Cooperative Control Layer

The final layer in the HUAS network architecture consists of either semi-autonomous (i.e. operator in the loop), autonomous, or cooperative control algorithms. These algorithms utilize the application layer networking software to coordinate aircraft and provide sensor and telemetry data to subscribers, including other aircraft and multiple dispersed users. It is at this layer that the CSAT algorithm is implemented.

## IV. RESULTS

#### A. HWIL MAV Robot Control

In order to facilitate development and testing of the HUAS systems, a hardware-in-the-loop (HWIL) simulation capability has been created. This laboratory environment uses all the hardware and software components of the full HUAS system with the exception of the autopilot sensor measurements. Instead, modified sensor measurements are given as inputs to the navigation subsystem from flight simulator software provided with the PiccoloPlus or from internal simulation components of the CU MAVs. The navigation subsystems of each autopilot output the control surface commands to software components that simulate the aircraft flight. The result of the HWIL system is that the UA "thinks" it is flying while sitting on the lab bench, including deflection of the aerodynamic control surfaces and actuation of engine throttle. In this way, system interactions can be tested under simulated flight conditions; the multi-tiered network can be validated under benign conditions; and coordination of multiple heterogeneous vehicles can be demonstrated inside the laboratory.

Using the HWIL setup several simulations have been conducted to demonstrate various components of the CSAT algorithm. Cooperative tracking between 3 Ares UA of a single target has been demonstrated HWIL [5]. This simulation did not require any user input or a ground station. Each UA used the telemetry stream from the other two UA to maintain phasing while orbiting the target.

Following successful simulation of SAV ground vehicle tracking, this software was simplified and broken apart to provide tracking functionality for the MAVs. The gateway to the MAV network (whether a ground station or Ares MS) maintains the coordination and assignment functionality, while the MAVs implement the functionality required to allow control of the UA via inputs defined in Equation 1.

Figure 4 shows the results from the HWIL simulation of two MAVs assigned to track a moving ground node. The trajectory of the target node is in black, and the trajectories of the two MAVs are in green and red. The assignment and coordination in this case took place on a groundstation laptop, but could have just as easily been performed by a Soekris SBC in an Ares SAV.

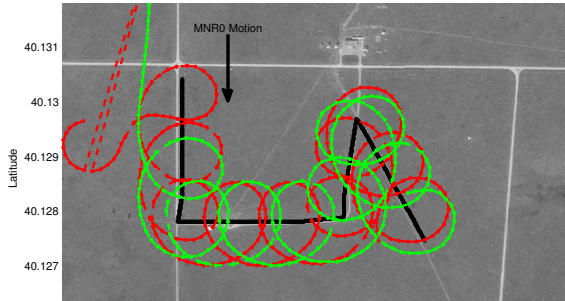


Fig. 4. HWIL showing two MAVs tracking a moving ground target.

Given the limited amount of available hardware, HWIL simulations involving more than a handful of aircraft was not possible. Software simulations were therefore required to demonstrate CSAT using a larger number of vehicles. An simulation was run using 9 aerial robots searching an area that contains 4 moving targets. The aerial robots move with a nominal airspeed of  $v_0 = 20$  m/s, have a maximum turn rate  $\omega_{max} = 0.2$  rad/sec, and are allowed deviations of  $\Delta v = 5$  m/s for coordination. The environment is 5000 meters wide and 8000 meters long. Initial positions of the 9 aircraft and 4 targets are randomly selected. Ground targets move with constant velocity with speed randomly chosen on the interval  $[4, 10]$  and random initial headings. Targets change direction only upon reaching the boundary of the environment.

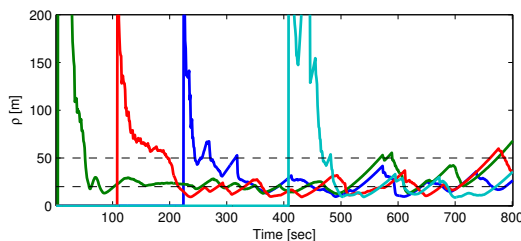


Fig. 5. Square root of the maximum singular value of the position error covariance matrix versus time.

The task assignment algorithm attempts to keep the position error  $\rho_k$  between 20 and 50 meters. Each active target is assigned to two robots who track the target until  $\rho \leq 20$ m and all remaining robots are assigned to search. The approximate average robot and target error velocities are set to  $\bar{v} = 0.8v_0$  and  $\bar{p} = 2\sqrt{\rho(\mathbf{P}_v)}$ , respectively. Centralized task assignment is performed using binary integer programming and the coverage map described in [3].

Figure 5 shows a plot of  $\rho$  for each target as a function of time. The dashed lines denote the upper and lower bounds

of the desired position error. The CSAT algorithm is able to provide significant coverage of the environment while also keeping the target uncertainty below the specified bound most of the time (Fig. 5). Since the approximations  $\bar{v}$  and  $\bar{p}$  were set heuristically, they are not always conservative and the target uncertainty occasionally grows above the given limit.

## V. CONCLUSION

This paper presented a hierarchical control architecture that enables cooperative surveillance by a heterogeneous aerial robot network comprised of mothership unmanned aircraft and daughtership micro air vehicles. A new integrated search and tracking assignment algorithm is used for task assignment by the mothership vehicles while daughtership control uses a simple guidance vector field motion primitive for loiter and tracking tasks. A heterogeneous unmanned aircraft system is described that will be used to verify the control approach described here.

Current work focuses on demonstrating the entire CSAT architecture using the hardware-in-the-loop simulation capability of the experimental platform. Implementation of the DS task assignment algorithm is currently being done. Future work will address distributed negotiation between MS vehicles to claim tasks that are then assigned to their DS vehicles. Also, experimental validation of all components of the CSAT system are underway.

## REFERENCES

- [1] D. Gross, S. Rasmussen, P. Chandler, and G. Feitshans, "Cooperative operations in urban terrain (counter)," in *Defense Transformation and Network-Centric Systems*. Edited by Suresh, Raja. *Proceedings of the SPIE, Volume 6249*, 2006.
- [2] D. Ratner and P. J. McKerrow, "Aerial tethered robotics system with hovering-hopping agents for security and rescue operations," in *Proc. of AUVSI 2006 INTL Conference on Unmanned Systems North America*, Orlando, Canada, Aug. 2006.
- [3] E. W. Frew and J. Elston, "Target assignment for integrated search and tracking by active robot networks," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, 2008.
- [4] E. W. Frew, "Cooperative stand-off tracking of uncertain moving targets using active robot networks," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy, May 2007.
- [5] J. Elston and E. W. Frew, "Net-centric cooperative tracking of moving targets," in *AIAA Infotech@Aerospace*, Rohnert Park, CA, May 2007.
- [6] W. Pisano, D. Lawrence, and P. Gray, "Autonomous uav control using a 3-sensor autopilot," in *AIAA Infotech@Aerospace 2007 Conference and Exhibit*, May 7-10 2007.
- [7] T. Brown, B. Argrow, E. Frew, C. Dixon, D. Henkel, J. Elston, and H. Gates, "Experiments using small unmanned aircraft to augment a mobile ad hoc network," in *Emerging Technologies in Wireless LANs: Theory, Design, and Deployment*, B. Bing, Ed., 2007, ch. 28, pp. 123-145.
- [8] D. Lawrence, K. Mohseni, and R. Han, "Information energy for sensor-reactive uav flock control," in *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop, and Exhibit*, Sept. 20 - 23 2004.
- [9] J. Elston, E. W. Frew, and B. Argrow, "Networked uav communication, command, and control," in *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, August 2006.