Proceedings of the 2007 IEEE Symposium on Computational
Intelligence in Security and Defense Applications (CISDA 2007)

An Intelligent Controller for Collaborative Unmanned Air Vehicles

Gregory L. Sinsley*, Jodi A. Miller†, Lyle N. Long,
Brian R. Geiger, Albert F. Niessner, Jr., and Joseph F. Horn
The Pennsylvania State University, University Park, PA 16802

*Email: gls189@psu.edu
†Email: jodi@psu.edu, Telephone: (814) 863-4077

Abstract— **This paper describes an implementation of an autonomous Intelligent Controller (IC) architecture for collaborative control of multiple unmanned aerial vehicles (UAVs). Collaborative capabilities include formation flying, search of an area, and cooperative investigation of a target. The IC provides capabilities for sensor data fusion, internal representation of the real-world, and autonomous decision making based on the IC's world model and mission goals. Results of flight tests demonstrating these capabilities are presented. Future work, such as integration of different sensors and collaboration with heterogeneous vehicles, is discussed.**

I. Introduction

Military Unmanned Aerial Vehicles (UAVs) have proved to be quite capable in performing tasks that are less desirable for manned aircraft. UAVs such as Global Hawk and Predator are currently being used for reconnaissance and surveillance in various global conflicts. In the Department of Defense roadmap for Unmanned Aerial Vehicles [1], lower downside risk and higher confidence of success are listed as two motivators for the continued development of unmanned air systems.

One way to greatly increase the autonomy of UAVs is to use intelligent control technologies. There are many different approaches to intelligent control. A few of them are described in references [2]–[5]. There is no universal consensus on how to define or measure an intelligent system. Evans and Messina [6] list several characteristics an intelligent system can have. These include: adaptability, learning capability, nonlinearity, autonomous symbol interpretation, and goal-oriented and knowledge-based behaviors. The Intelligent Controller (IC) described in this paper is based on the Intelligent Controller architecture developed at the Applied Research Laboratory/Penn State (ARL/PSU) [7]–[11].

This paper describes how ARL/PSU's Intelligent Controller is applied to the control of UAVs. The Penn State UAV Lab employs several modified SIG Kadet Senior model aircraft flying under autonomous control (Fig. 1). Table I lists the specifications of the UAVs. The UAVs use the Piccolo Plus autopilot developed by Cloud Cap Technology [12]. Each UAV also carries an onboard computer running the IC software. The IC is responsible for high level mission planning and coordination among UAVs. For complete details on our UAV testbed see [13].

Fig. 1. SIG Kadet UAVs

TABLE I
Aircraft Specifications

Wingspan	80 inches
Wing Area	1180 sq. inches
Length	64 3/4 inches
Empty Weight	6 1/2 pounds
Gross Weight	14 pounds
Wing Loading	1.7 pounds/ft^2
Engine	0.91 cubic inch 4-stroke

Several new features have been added to the UAV Intelligent Controller in the past year, such as image processing and path planning capabilities. Features previously tested in simulation have now been tested in-flight. This paper describes the IC's new features and gives details of the flight tests.

II. ARL/PSU's Intelligent Controller Architecture

Fig. 2 is a high level illustration of ARL/PSU's Intelligent Controller. The IC architecture is composed of two main modules: Perception and Response. The Perception module is where input data is analyzed, fused and interpreted to create an internal representation of the external world. Using the world view generated in Perception, the Response module performs situation assessment, mission planning and re-planning, and carries out execution of the current plan. Output from Re-

1-4244-0700-1/07/$20.00 ©2007 IEEE

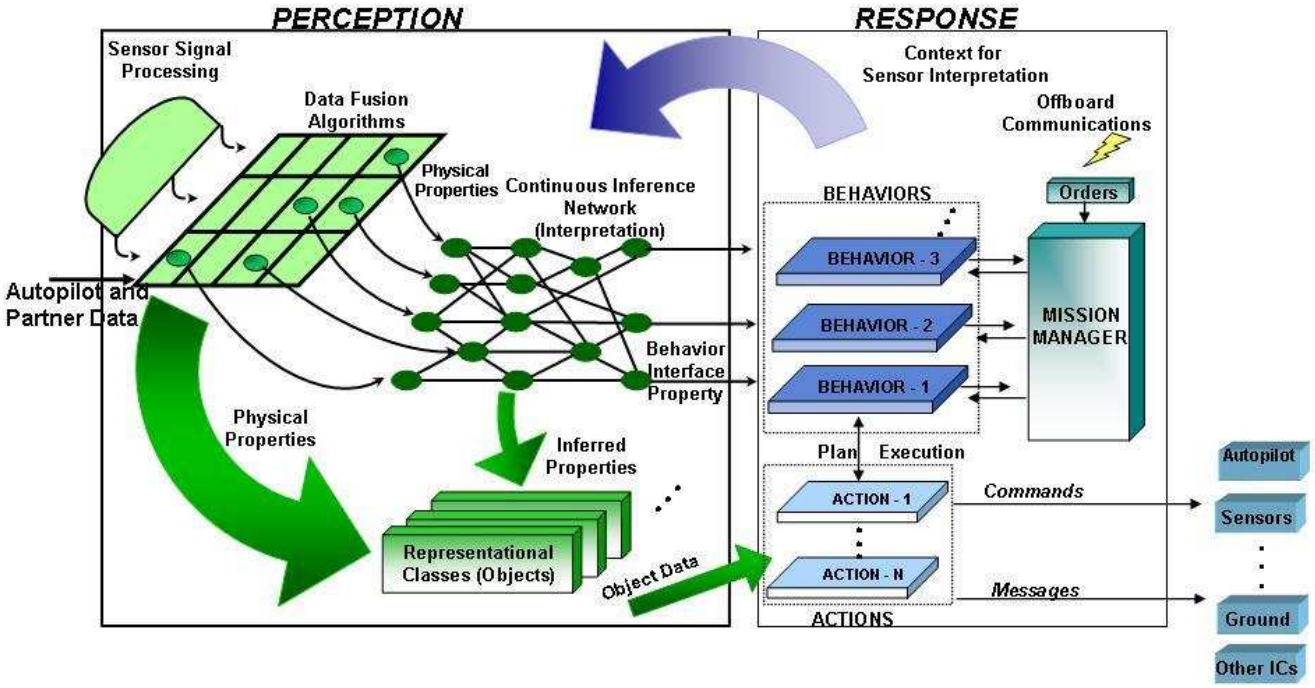

Fig. 2. ARL/PSU IC Architecture

sponse includes commands and communications to external systems and vehicle subsystems. These modules are described in detail below.

A. Perception

The Perception module creates the situational awareness for the IC. It builds internal representations of the external world relevant to the IC using sensor data and other messages as inputs. These internal representations are called Representational Classes (RCs). The Perception module makes inferences and recognizes the existence of properties in its RCs from input data, which may be incomplete and potentially erroneous. Data fusion algorithms and Continuous Inference Networks (CINets) are used [7]–[9] to update Perception's real world model. The data fusion algorithms are responsible for fusing new sensor data so that the existing RCs can be updated. The CINets infer properties or events, such as ”target”, by appropriately combining multiple pieces of information in an automatic recognition process.

An Input Interface module converts data streams external to the IC into forms required by the Perception module. It should be noted that some amount of sensor signal processing is typically performed on raw sensor data before the Input Interface receives it. Data is accumulated in a buffer and released to Perception at discrete time intervals referred to as *processing cycles*. A processing cycle is the amount of time required for an effector to complete a command and return data to the IC. Alternatively, a timer can determine the processing cycle. Typically, the processing cycle is on the order of one second, the human control frequency, but it can be any value required by the application and supportable by the speed of the processors involved.

B. Response

The Response module plans and executes *in real time* a plan of action to perform a specific mission, given the situational awareness derived by the Perception module. The Response module is composed of a number of autonomous Behaviors (agents). Each Behavior acts independently, generating and executing plans in real time, monitoring plan progress, and adapting plan execution as appropriate. A Mission Manager is used to arbitrate between Behaviors requesting control.

A Behavior monitors Perception for the existence of objects in its interest as indicated by certain high-level inferred properties (e.g., an “attack” behavior would look for the presence of objects classified as “targets”). When a Behavior detects such an object, it notifies the Mission Manager that it is ready to take control. The Mission Manager possesses a definition of the current mission and uses it to determine the relative priority of each Behavior at any point during the mission. It selects one or more Behaviors that are requesting to be enabled and turns operations over to each, as appropriate. The Mission Manager may also grant certain Behaviors control without a specific request from the Behavior. This occurs when the Behavior does not depend on the existence of certain types of objects before it can function.

Each Behavior has one or more Actions that generate its output commands and messages. These commands and messages are the output of the Response module. Commands are sent to effector subsystems, such as commands to an autopilot for vehicle control or commands to sensor controllers to configure sensors to gather needed information. Other outputs include orders, queries, or advisories to other ICs or humans in the team.

III. UAV Intelligent Controller (IC)

The role of the IC in the UAV is to provide mission control functionality for the aircraft. Each UAV runs an identical copy of the IC on its onboard processor. Properties of the specific vehicle that the IC is contained in (such as vehicle ID number, IP address, and sensors used) are downloaded in configuration files at mission startup.

The UAV IC includes capabilities for autonomous operations for individual units as well as collaboration between UAVs (and ultimately other heterogeneous autonomous vehicles). These capabilities include: Standby, Communicate, Flight Path, Coordinate Investigate, Investigate, and Follower. Each of these operations is implemented as an independent Behavior as described in the previous section. The UAV IC is discussed in detail in the following sections.

A. UAV IC Perception Module

The Perception Module is where the IC's world view is internalized and stored in the form of Representational Classes (RCs). An IC's representation of the UAV it is embedded in is contained in a Self RC. The UAV IC Input Interface receives vehicle information, which includes telemetry, control, and commands data from the Piccolo Plus Autopilot, and reformats the data in a form required for the Perception Module. Similarly, status information from partner UAVs is stored in a Partner RC. The UAVs use an 802.11b ad-hoc network to communicate with each other and the ground. Orders from the ground are stored as Standing Orders and require no Perception processing, i.e. no data fusion and inferencing.

Because the IC utilizes processed sensor data, image processing from a camera or other visual sensor is performed by a separate piece of software residing on the same (or a separate) onboard processor. The image processing program utilizes both Intel's OpenCV library [14] and code developed in-house. OpenCV implements many common computer vision routines, optimized for use on Intel processors. Using routines from OpenCV, we have written a program that can find red or blue blobs in an image, and triangulate their location using a triangulation algorithm that utilizes telemetry from the autopilot. A similar approach can be used to find the location of objects of a certain shape (such as an 'X', a rectangle, or a bull's eye). We are currently working on extending the algorithm to be able to determine the relative velocity of the target so that moving objects can be tracked.

Once the software finishes processing an image, relevant information (such as location, velocity, shape, size, color, brightness, etc.) is sent in report form via TCP/IP to the IC. When the IC receives the report, it broadcasts it to its partner UAVs. The IC receiving the original report uses CINets [7]–[9] to infer if the object detected is an "object of interest", i.e., an object worth investigating further. CINets infer confidence factors (values between 0 and 1) for the set of inferred properties defined for a particular class. CINets are used as opposed to a binary mechanism because physical variables are usually continuous and this avoids loss of data. Object physical properties and confidence factors are stored in Track RCs.

B. UAV IC Response Module

The Behaviors and Actions developed for the current UAV Response module are shown in Fig. 3. The highest level of planning and control is the Mission Manager. It arbitrates when multiple behaviors request control. It decides which Behaviors take control each IC cycle (a pass through Perception and Response) based on a relative priority scheme determined by the given mission. The mission is either loaded at startup or transmitted during mission execution to modify the mission on-the-fly. The Actions generate the output commands and messages to vehicle subsystems and partner UAVs and are always executed when needed by a particular Behavior.

The priority scheme used by the Mission Manager is the same for our independent and collaborative missions. The Communicate and Coordinate Investigate Behaviors will always be granted control when they request it. Because these Behaviors involve communications only and do not control the aircraft, there is no conflict of resources and both can operate in conjunction with Behaviors that do control the vehicle. For those other Behaviors, the priority order is: Standby, Investigate, and Flight Path. The Behaviors are described below in detail.

1) Standby: Standby is the default behavior of the UAV IC. In Standby, the UAV will fly the base flight plan as given to the autopilot by a human on the ground before the IC takes control (and provided to the IC at startup). The IC will not send any commands to the autopilot when this Behavior is in control. The IC can also be ordered to go into Standby at any time by a human on the ground. When such an order appears and Standby requests to be enabled, the Mission Manager will always grant control to the Standby Behavior.

2) Communicate: The Communicate Behavior looks for incoming messages that require a response and sends out an appropriate reply. For example, if a Request For Status message is received from another UAV, the Communicate Behavior will reply with a Status message containing the UAV's ID number, location, altitude, speed, availability, and sensor capabilities. In response to a partner request for collaboration assistance (to investigate an object of interest), a message either accepting or declining the task is sent. If the IC accepts a request for assistance, it will set a state in the appropriate track indicating this UAV needs to investigate the object. Because the Communicate Behavior can operate in conjunction with other Behaviors, it is always granted control by the Mission Manager when requested.

3) Flight Path: The Flight Path behavior will request control when a Flight Path Order is the current standing order. When the Flight Path Behavior is enabled, it sends a user defined flight plan to the autopilot and monitors waypoint

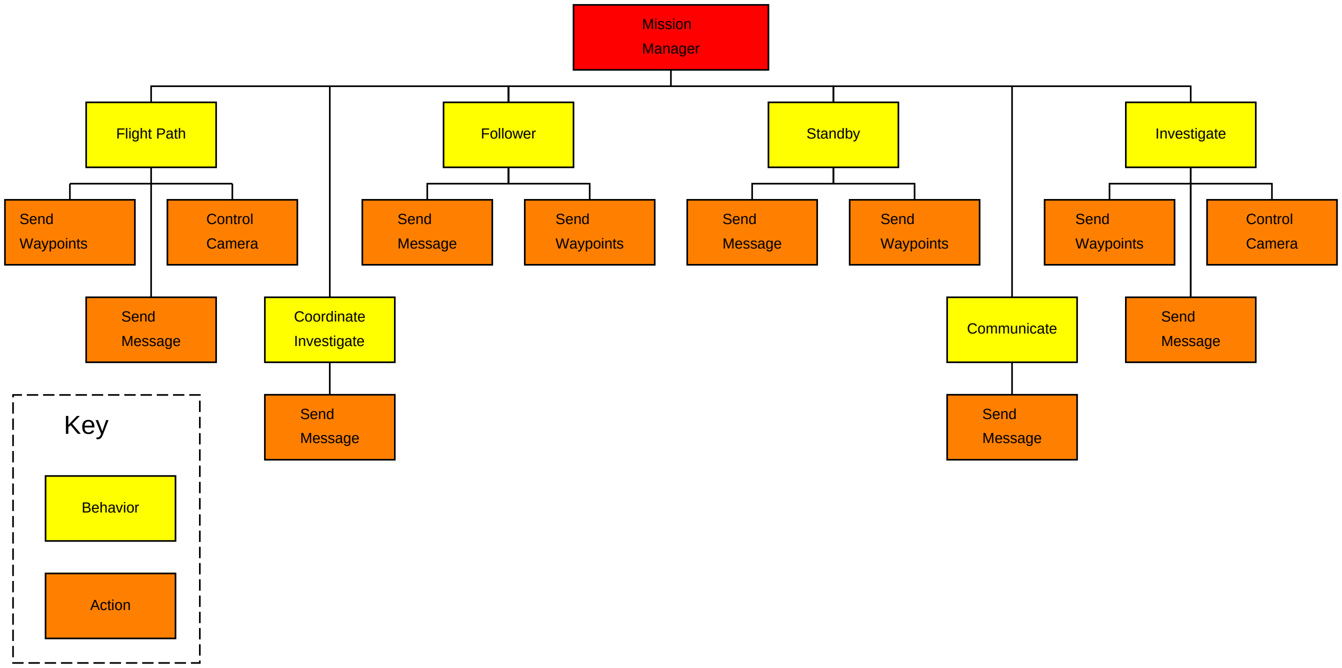

Fig. 3. UAV IC Response Module

progress. The IC will remain in this Behavior until another order is sent from ground or an investigation occurs.

4) Coordinate Investigate: The Coordinate Investigate Behavior is used to coordinate partners in investigating objects of interest. It utilizes a negotiation scheme to help determine, select, and request assistance from partner UAVs. The Coordinate Investigate Behavior requests to be enabled when the UAV itself detects an object that is classified as an "object of interest" from its own visual sensor.

Negotiations between partners to collaborate on investigating an "object of interest" involve a series of message exchanges, culminating in one (or no) UAV going to investigate the object. The simplest exchange, where there are no communication drop-outs, occurs as follows:

1) UAV 1 receives contact report from onboard visual sensor and broadcasts the report to all partner UAVs. UAV 1 determines track is an "object of interest."
2) UAV 1 initiates negotiations to coordinate an investigation by broadcasting a Request For Status message to all partner UAVs and waits for responses.
3) Partner UAVs send out Status messages, indicating current position, speed, availability, and visual sensor capability.
4) UAV 1 determines best partner to collaborate with as a function of partner location, availability and visual sensor capability. UAV 1 sends best partner a message requesting assistance.
5) Best partner responds to UAV 1 either accepting or declining the task.
6) If the best partner accepts the task, negotiations are completed. If the best partner declines the task, UAV 1 tries again with the next best partner (which may turn out to be itself or no UAV).

5) Investigate: The Investigate Behavior looks for a track that has been marked for the UAV to investigate (as a result of investigation coordination). If such a track exists, the Behavior will request control from the Mission Manager. Upon activation, it will send the autopilot an "investigate speed" and the coordinates of the object of interest. The Investigate Behavior then monitors the progress of the UAV as it approaches the object. When the object is reached, the IC will send a command to activate the UAV's visual sensor and orbit the target for a predefined amount of time. When the UAV finishes orbiting, it will broadcast a message to all its partners notifying them that the investigation is complete. If there are no other objects to investigate, the Mission Manager will return control to the previous behavior, and the UAV will resume its previous flight plan and speed.

The Path Planner is a separate process that may be used by the Investigate Behavior. Instead of sending a destination waypoint to the autopilot, which then determines the path to the object of interest, Path Planner generates its own optimal path that ensures the object will remain in the camera's field of view for the greatest percentage of time [15]. One goal for using Path Planner is to be able to track multiple targets that are either stationary or moving. Another goal is to use it with multiple UAVs so that through a combination of observations by sensors aboard different UAVs, the target would be observed for a greater percentage of the time than with a single UAV. In the case of a single target, two UAVs will provide nearly continuous sensor coverage.

6) Follower: The Follower Behavior allows one UAV to follow another at a constant distance above (or below) and behind. Currently, the follower UAV flies the same set of waypoints as the leader UAV. The only variable is the follower's speed. The follower uses PID feedback control in order to stay at a constant distance behind the leader. Eventually, the need to know the leader's path will be eliminated. Instead, the follower will generate its own waypoints based upon the GPS location of the leader. We are also exploring other ways to determine the leader's position, such as triangulation or optical flow. (This could be useful in a battlefield situation, where you would want radio silence.) Applications of this Behavior include formation flying and mid-air refueling.

IV. Flight test results

Features of the UAV Intelligent Controller have been tested both independently and together in the air using our SIG Kadet Senior UAVs. This section describes these tests.

A. Image processing

The image processing algorithm was tested in two stages. In the first stage, the basic algorithm was tested on the ground. A webcam was mounted on a tripod, and its position and orientation were measured with a tape measure, compass, and protractor. The position of a red object was measured similarly. Two images of the object were taken from different camera locations. The algorithm determined the position of the object through triangulation and was found to be accurate to within 5% error (expressed in terms of distance from the camera).

In the second stage of testing, images of objects on the ground were taken using both a webcam and a digital camera from a UAV in-flight. (The webcam sends video to a ground base station via a 2.4 GHz link in real time, while the digital camera stores images on a memory card for latter retrieval on the ground.) After the flight, these pictures and the UAV's telemetry information were uploaded onto a PC. Using this data, the image processing program triangulated the location of objects (barrels, posters, cars, etc.) on the ground. These locations were then compared to the locations of these same objects measured using a handheld GPS unit, and found to be accurate to within 10^{-7} radians (less than a meter). Fig. 4 shows the location of a target and 3 estimates of the target's location.

B. Path planning

The Path Planner software was tested independently of the UAV IC [15]. It was executed offline to generate a path that allowed maximum sensor coverage time for investigating an object at a known location. The planner accounts for sensor properties, UAV performance limits, and wind speed. The generated path was then used in flight by a path following algorithm. The path follower sends turn rate commands to the autopilot to follow the path. The next stages in this area of research will be to convert the planner to online operation and incorporate it into the IC Investigate Behavior.

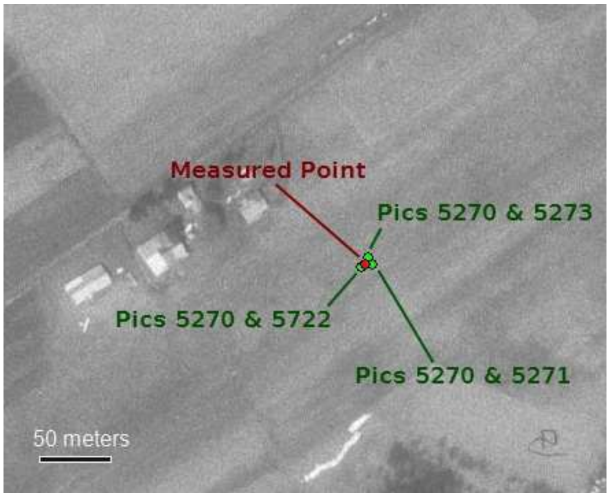

Fig. 4. Estimated and actual location of a target

C. Collaboration

The purpose of this flight test was to demonstrate the collaborative capabilities of the UAV IC. Two UAVs were placed in their base flight plans while under autopilot control only. (For safety reasons, the two planes were flown at different altitudes.) Shortly thereafter, the IC on each UAV was started, providing mission control for the aircraft. Each UAV IC was subsequently sent a Flight Path Order, which contained a new flight plan. The Flight Path Behavior of each IC took control as expected and the UAVs left the base flight plan for the ordered flight plan.

While in its new flight plan, UAV 1 received a camera report for an object of interest and broadcasted the report to UAV 2. UAV 1 entered the Coordinate Investigate Behavior (Flight Path still remained active) to determine, and negotiate with, the best partner to investigate the object. UAV 2 was determined to be the best partner and a series of message exchanges between the two aircraft occurred (as described in Section III-B.4), resulting in UAV 2 accepting the task of further investigating the object of interest.

UAV 1's Coordinate Investigate Behavior completed and UAV 1 continued to fly its ordered flight plan. UAV 2 entered the Investigate Behavior (Flight Path was interrupted at this time), deviated from its current flight plan to fly to the object, enabled its visual sensor and orbited the object for ninety seconds. Upon completion of the investigation, UAV 2 returned to its previously ordered flight plan and the Flight Path Behavior resumed control. Fig. 5 is a photograph taken during this test.

V. Future work

There are several upgrades to the system that are planned for the future. The first priority is to optimize the path planner algorithms so that they can be run in real time as part of the UAV IC. Next, the intent is to streamline the image processing algorithms so that they can provide results to the UAV IC in

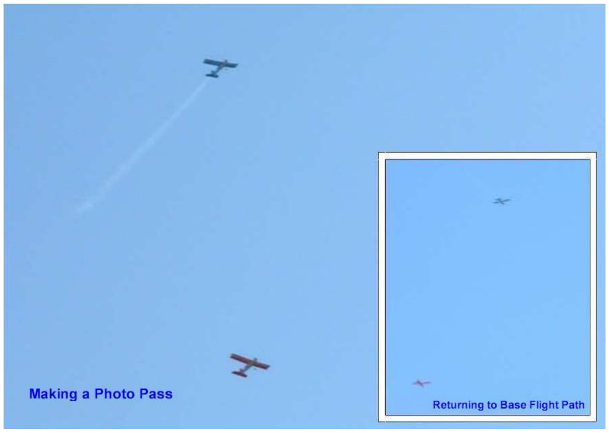

Fig. 5. Photograph of IC demo flight

a more timely manner.

Another step will be the integration of more sensors. One important sensor is a gimbaled camera. A major limitation of the current camera is that it only takes pictures of objects directly below the UAV, causing objects to go out of the field of view of the camera while the UAV is in a banked turn. A gimbaled camera will allow the UAV to keep an object in its field of view for a much greater percentage of time. Another new sensor that will be added is a magnetometer to provide better estimates of the UAV's heading. This will increase the accuracy of triangulation results for image processing. A magnetometer should be easily integrated because the autopilot is equipped with built-in capabilities to interface with one.

The ARL/PSU Intelligent Controller architecture is applicable to the design of an intelligent controller for any autonomous vehicle. Many possibilities exist for air to ground vehicle collaboration. For example, a UAV with a Follower Behavior could follow a vehicle or a convoy of vehicles on the ground. Another application would be an investigation involving both air and ground vehicles. An object detected from the air by a UAV could also be investigated by a ground vehicle at the UAV's request, yielding better data about the object than the UAV alone could provide.

VI. Conclusions

The ARL/PSU Intelligent Controller architecture has been successfully applied to the design of an Intelligent Controller (IC) for UAVs. This IC provides mission control for individual aircraft as well as collaborative control capabilities for multiple vehicles. It is easily expandable to incorporate new functionality as desired. Using the UAV flight system developed by researchers at both ARL/PSU and the Penn State Department of Aerospace Engineering [13], successful flight tests have been conducted demonstrating current UAV IC capabilities.

Acknowledgment

The authors would like to acknowledge James Ross for his help during the flight test portion of the work.

References

[1] "Unmanned aircraft systems roadmap," Office of the Secretary of Defense, Washington, D.C., Aug. 2005. [Online]. Available: http://www.acq.osd.mil/uas/

[2] D. A. White and D. A. Sofge, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*. Van Nostrand Reinhold, 1992.

[3] A. M. Meystel and J. Albus, *Intelligent Systems: Architecture, Design, Control*. Wiley, 2001.

[4] G. A. Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control*. Cambridge: MIT Press, 2005.

[5] L. N. Long, S. D. Hanford, O. Janrathitikarn, G. L. Sinsley, and J. A. Miller, "A review of intelligent systems software for autonomous vehicles," to be presented at the IEEE Computational Intelligence for Security and Defense Applications Conference, Honolulu, Hawaii, Apr. 1-5 2007.

[6] Evans and E. Messina, "Performance metrics for intelligent systems," in *Proceedings of the 2000 PerMIS Workshop*, Gaithersburg, MD, Aug. 14-16, 2000, pp. 101–104.

[7] J. A. Stover *et al.*, "Continuous inference networks for autonomous systems," in *IEEE Conference on Neural Networks for Ocean Engineering*, Aug. 17 1991, pp. 177–183.

[8] J. A. Stover and R. E. Gibson, "Modeling confusion for autonomous systems," *SPIE, Science Artificial Neural Networks*, vol. 1710, pp. 547–555, 1992.

[9] J. A. Stover, D. L. Hall, and R. E. Gibson, "A fuzzy-logic architecture for autonomous multisensor data fusion," *IEEE Trans. Ind. Electron.*, vol. 43, pp. 403–410, 1996.

[10] J. A. Stover and R. E. Gibson, "Controller for autonomous device," U.S. Patent 5,642,467, June, 1997.

[11] J. A. Stover and R. Kumar, "A behavior-based architecture for the design of intelligent controllers for autonomous systems," in *IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics*, Cambridge, MA, Sept. 15-17 1999, pp. 308–313.

[12] [Online]. Available: http://www.cloudcaptech.com

[13] J. Miller, P. Minear, A. Niessner, A. DeLullo, B. Geiger, L. Long, and J. Horn, "Intelligent unmanned air vehicle flight systems," in *AIAA InfoTech@Aerospace Conference*, Washington D.C., Sept. 26-29 2005, paper 2005-7081, (to appear in the Journal of Aerospace Computing, Information, and Communication, 2007).

[14] [Online]. Available: http://www.intel.com/technology/computing/opencv/

[15] B. R. Geiger, J. F. Horn, A. M. DeLullo, L. N. Long, and A. F. Niessner, "Optimal path planning of UAVs using direct collocation with nonlinear programming," in *AIAA Guidance, Navigation, and Control Conference*, Keystone, Colorado, Aug. 21-24 2006, paper 2006-6199.