AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit
20 - 23 September 2004, Chicago, Illinois

AIAA 2004-6490

Flight Demonstrations of Cooperative Control for UAV Teams

Jonathan How[*], Ellis King[†], and Yoshiaki Kuwata[‡]
Aerospace Controls Laboratory
Massachusetts Institute of Technology

This paper discusses the development and testing of a unique testbed consisting of a fleet of eight autonomous unmanned aerial vehicles (UAVs) that was designed as a platform for evaluating autonomous coordination and control algorithms. Future UAV teams will have to autonomously demonstrate cooperative behaviors in dynamic and uncertain environments, and this testbed can be used to compare various control approaches to accomplish these coordinated missions. A hierarchical configuration of task assignment, trajectory design, and low-level, waypoint following, are used in a receding horizon framework to control the UAV team. Numerous trajectory optimization and team coordination algorithms have recently been developed to execute these UAV missions. This paper highlights several of these algorithms and presents typical results for representative experiments. These demonstrations of the high-level planning algorithms on scaled vehicles operating in uncertain and dynamic environments represent key steps towards transitioning them to future UAV missions.

I. Introduction

UAVs offer advantages over conventional manned vehicles in many applications because they can be used in situations otherwise too dangerous for manned vehicles (e.g., eliminating anti-aircraft defenses) and without being weighed down by the systems required by a pilot, UAVs are capable of staying aloft on longer surveillance missions.[1] While the roles and capabilities of UAVs are growing, current UAV control structures were conceived with limited roles in mind for these vehicles. Thus it is necessary to improve on this control structure in order to fully exploit the expanding capabilities of UAVs. This includes developing techniques to optimize the coordination of a fleet of UAVs, which is comprised of the coupled subproblems of determining sub-team composition, allocating resources, and optimizing vehicle trajectories.[2] These are all computationally intensive optimization problems that require good situational awareness to achieve coordinated and cooperative behaviors. Numerous algorithms have recently been developed to achieve this cooperative behavior,[3,4] but a key step towards transitioning these high-level algorithms to future missions is to successfully demonstrate that they can handle the implementation challenges using scaled vehicles operating in realistic environments. Thus two testbeds have been developed at MIT to perform these demonstrations. One uses multiple rovers and blimps operated indoors to emulate a heterogeneous fleet of vehicles performing a search and rescue mission. The second uses eight UAVs that are flown autonomously using a commercially available autopilot from Cloud Cap Technology. Performing experiments on these testbeds will highlight the fundamental challenges associated with: (i) planning for a large team in real-time; (ii) developing controllers that are robust to uncertainty in situational awareness, and are sufficiently flexible to respond to dynamic changes; and (iii) using communication networks and distributed processing to develop integrated and cooperative plans.

[*] Associate Professor, MIT Department of Aeronautics and Astronautics, Senior Member AIAA. jhow@mit.edu
[†] Research Assistant, MIT Department of Aeronautics and Astronautics. etking@mit.edu
[‡] Research Assistant, MIT Department of Aeronautics and Astronautics. kuwata@mit.edu

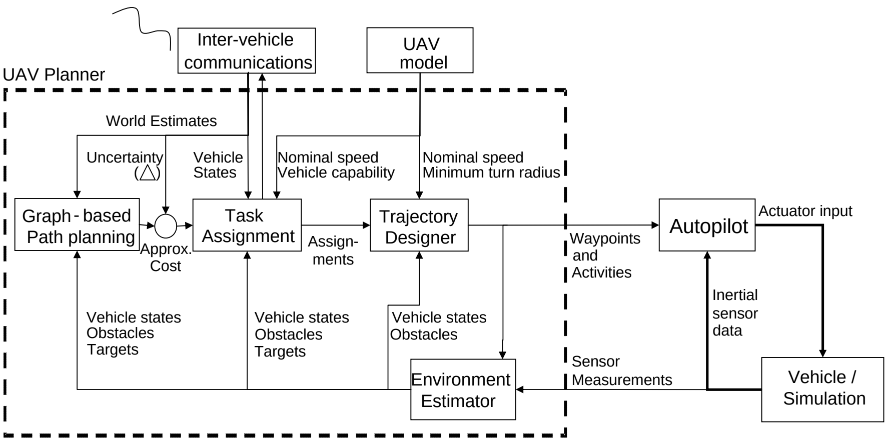

Figure 1. System algorithm architecture for the UAV testbed.

II. Coordination Algorithms

Mixed-integer Linear Programming (MILP) has previously been shown to provide a natural framework for posing coordination problems, and approximations such as the *decomposition approach* have proven to provide accurate yet tractable solutions to the overall problem.[4,5,6] The decomposition approach simplifies the coupling between the assignment and trajectory design problems by calculating and communicating only the key information that connects them. This is achieved using an approximate cost-to-go calculation to obtain good estimates of the costs associated with feasible paths around "obstacles" (e.g. buildings, no-fly-zones) in the environment. These costs are then used in the assignment problem solved using the *petal* algorithm.[6,7] Uncertainty in the target classification due to poor or conflicting information enters the problem as uncertainty in the assignment costs. As demonstrated in Ref. 8, the assignment process must be robust to these types of uncertainty, and it is also vital to ensure that the reconnaissance and strike tasks are allocated simultaneously to provide the most benefit to the strike part of the missions.

Task Assignment – While the decomposition approach greatly reduces complexity, the problem of task assignment with precedence constraints has been shown to be NP-Hard, and achieving the exact solution for a large team of UAVs is computationally intensive and not suitable for real-time applications. The petal algorithm uses a heuristic method to prune out the solutions that are not likely to be part of the optimal solution, which significantly speeds up the task assignment process. While this approach has been shown to work well for small problems, it is still difficult to run in real-time for problems with larger numbers of vehicles and tasks. To perform reassignment in real-time as required in a dynamic, frequently changing environment, we have extended the petal method to develop a receding horizon task assignment (RHTA) algorithm.[9] RHTA significantly reduces the computation time by selecting at most m (typically less than 3) tasks for each UAV, then repeating the optimization over several iterations.[10] The process selects tasks to the mission list for each UAV, updates the UAV's position and time, removes the assigned tasks from the task list, and repeats until all the tasks are assigned. Timing and precedence constraints can be imposed using the approach in Ref. 6 or by simply requiring that tasks be removed from the list until all precedents have been assigned. The first of these two approaches is more complicated, but should yield a less conservative approach than the second.

Trajectory Optimization – The final step is to compute detailed UAV trajectories around the obstacles, which can be solved using a MILP-based receding horizon planner.[11] This approach has been proven to guarantee the arrival at the target in bounded time, as RH-MILP (or RH-TRAJ) uses a detailed trajectory model in the near term and an approximate path in the long term. This combination gives a good estimate of the cost-to-go and greatly reduces the computational effort required to design the complete trajectory. Discrepancies in the assumptions made in the two models are handled by ensuring that the planning horizon is sufficiently long.[12] Novel pruning and graph search algorithms have recently been integrated with RH-MILP, and these also have the effect of significantly reducing the computational load. In the following experimental sections, the trajectories shown are designed in real-time using this approach.

Control Architecture – Figure 1 shows the control architecture used for the UAVs.[4,6] Low-level control and the basic estimation tasks are run onboard, and the planning for the vehicles is done off-board. The *planner* outputs dynamically feasible waypoint lists and actions (i.e. classify, strike, assess) to the vehicles, and monitors the uncertain states of the vehicles and the world map. When significant changes to the situational awareness are detected, the cost map is then updated, the tasks are re-assigned and/or the trajectories are redesigned.

Robust and Adaptive Planner – Wind represents a large disturbance source for teams of UAVs, and due to vehicle limitations, it is important to handle this type of disturbance appropriately at all levels of the control system to ensure that the tasks assigned and trajectories designed are feasible. Recent work has focused on a multi-level approach to account for the effects of these wind disturbances. As shown in Figure 2, the aircraft is influenced by wind disturbances that consist of averaged, static components ($\overline{W}$) and turbulence (δW) that can be modeled using a Dryden model. These winds limit the aircrafts ability to follow the desired flight plan and execute the tasks at the desired times. The effects of the wind are compensated for with changes to the three parts of the control system shown in Figure 2.

The LQG timing compensator modulates the vehicle flight speed to reduce the effect of wind variations (distance Δd and velocity Δv errors) on the plan execution. The key point here is that the vehicle has strict saturation limits on the airspeed, which constrains the maximum error that can be compensated for at this level. If very large timing errors accumulate, these must be compensated for at the trajectory design and task assignment levels, but these two high-level planners must also be robust to the errors introduced to the plan execution by the wind. This is accomplished in the trajectory design by using the wind estimates $\hat{W}_k$ to account for the gross motion of the atmosphere to design paths that are dynamically feasible and consistent with the timing assignments. The "wind-robust" task assignment algorithm (RHTA) also uses estimates of the static wind $\hat{W}_k$ and wind uncertainty (and estimation error) $\tilde{W}$ to choose assignments that are robust to the possible variations in the vehicle flight times. This algorithm also chooses the reference velocity required to enforce the loitering constraints within the vehicle capabilities.

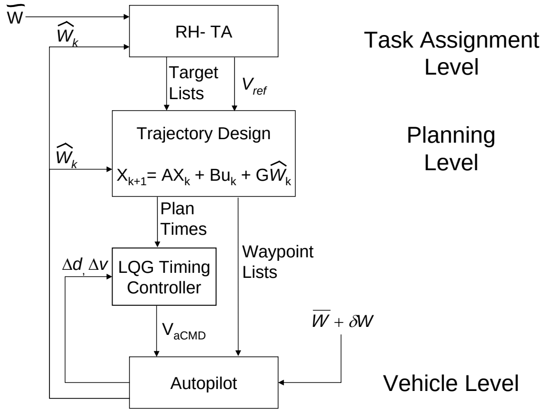

Figure 2. Multi-level estimation and timing planning/control approach used to compensate for environmental disturbances.

III. Testbed Infrastructure

The system infrastructure was set up to emulate a fully integrated fleet of UAVs, but maintain as much simplicity as possible in the vehicles themselves. All data passes through a central hub that performs data management between the planning computers and vehicles, effectively simulating communication delays, vehicle sensors and uncertainty in the environment. Using this setup greatly simplifies the testbed, while maintaining nearly all of the functionality of a fully integrated system. For example, as shown in Figures 3 and 4, the testbed can be used to investigate the impact of communication networking issues on the coordination problem by imposing various constraints on how the planning laptops communicate (using their own wireless or Ethernet links). Future work will demonstrate the effectiveness of various control architectures on the task assignment process, as would be seen in utilizing dynamic sub-teams of various compositions.

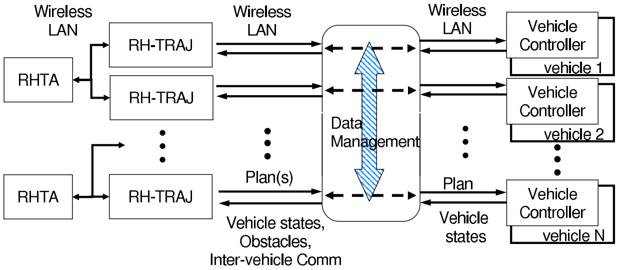

Figure 3. Control architecture: Decentralized path planning/Distributed task assignment.

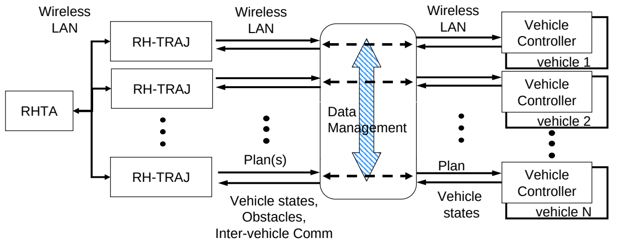

Figure 4. Control architecture: Decentralized path planning/Centralized task assignment.

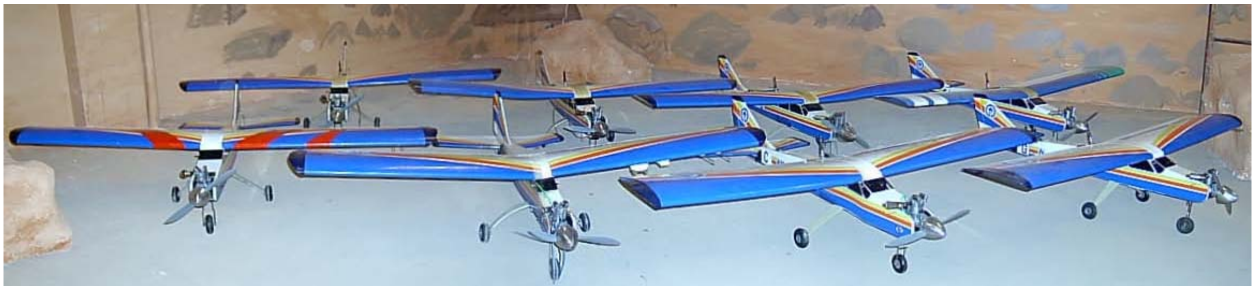

Figure 5. The fleet of 8 identical trainer ARF 60 aircraft used in the multi-UAV testbed at MIT.

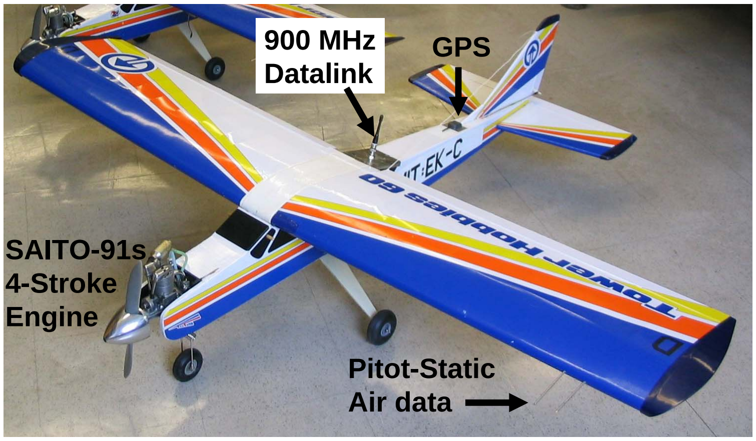

Figure 6. Trainer ARF 60 aircraft used in MIT multi-UAV testbed.

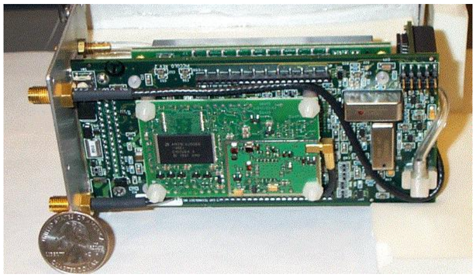

Figure 7. Commercially available Piccolo^{TM} Autopilot from Cloud Cap Technologies.

A key feature of the setup of this testbed is that much of the complexity of the system has been kept off-board the vehicles. This allows the aircraft to be scaled smaller than would be necessary if additional computers, batteries and other sensors were on board, yet the performance is similar because waypoint plans and high level control commands can be uplinked to the vehicles at a rate of up to 1Hz. Under the presumption that the low-level vehicle controllers are working well, the planning system would not need to respond faster, and in practice transmissions are made less frequently.

A. Tower Trainer ARF 60 Aircraft

In order to make successful demonstrations of multi-vehicle flights, the logistics require that the vehicles all have adequate minimum flight durations to ensure that there is sufficient time to perform the required ground operations. For a fleet of four vehicles, flight times greater than 40 minutes are needed in order to have sufficient time in the air to perform experiments. In addition, the vehicles must have sufficient wing loading capacity to carry additional weight from sensors and batteries. The vehicles selected for the testbed are commercially available Tower Trainer ARF 60 aircraft, which have easy handling characteristics and relatively large payload capacities. Only minor modifications are required to augment this class of aircraft to suit the mission requirements, which means that they can be quickly constructed and standardized across the entire fleet. In addition, maintenance and repairs are made much simpler by utilizing cheap, standardized aircraft for the fleet, and the logistics of flight tests are made much simpler by having vehicles with similar handling characteristics. Figure 5 shows the fleet of 8 trainer ARF 60 aircraft.

Property	Value
Wing Span	1.707m
Wing Area	0.520m^2
Chord Length	0.305m
Wing Incidence	1°
Wing Dihedral	5°
Gross Mass	5.26kg
Empty Mass	4.80kg

Table 1. Trainer 60 ARF aircraft parameters

A single ARF 60 aircraft is shown in Figure 6, with a list of important aircraft parameters in Table 1. The large wing area of the aircraft, combined with the four-stroke, Saito-91s (91 ccs) engine provides more than 3 lbs. of payload capacity, which is sufficient for the avionics, batteries, and additional sensors. An external fuel tank more than doubles the fuel capacity of the aircraft, which allows for extended flights of greater than 50 minutes with moderate throttle settings. The integration of GPS and air data sensors are minor modifications, providing the necessary measurements for autopilot control. The tower trainer aircraft

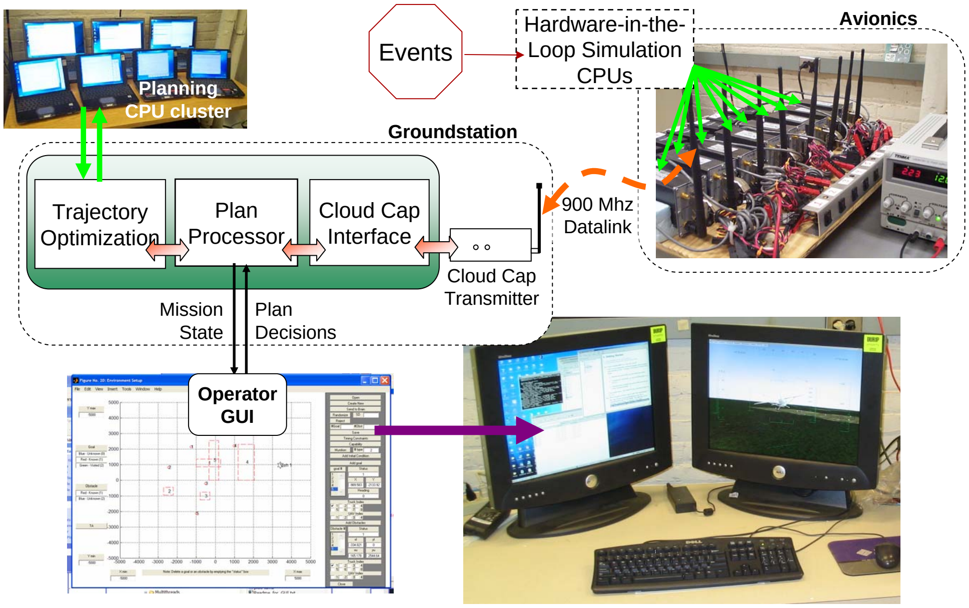

Figure 8. Hardware-in-the-loop configuration allows simultaneous simulation of 8 aircraft with the integrated planning system connected through the 912MHz data link, exactly as would be performed in flight.

are well suited for autopilot control because of their stable design for pilot training purposes. The stable configuration causes them to be less susceptible to upsets caused by turbulence, and the aircraft trim states are easily determined. However, because the aircraft is so stable, maneuverability is sacrificed. The reduction in performance, combined with minimum flight speeds of ≈20 m/s, dictate that slightly larger test areas be utilized to perform effective demonstrations, however this tradeoff makes sense for the proof-of-concept missions attempted in this phase of the project.

B. Autopilot

At the time the aircraft testbed was designed, the options for low-level vehicle control included either purchasing or constructing low-level autopilots. Due to proprietary source code restrictions, purchasing an autopilot would be less flexible to suit the requirements of the planning system, however the tradeoff in the time to complete the same tasks were significant[a]. As a result, the decision to purchase the equipment was made. After considering various options, the decision was made to use the Piccolo^{TM} autopilot from Cloud Cap Technology (Figure 7). This autopilot is used onboard the aircraft to perform the autonomous vehicle stabilization and waypoint navigation. Data is sent by one watt transmitter over the 912MHz datalink, which permits the vehicles to navigate up to three miles from the ground station. This link can also be used to continuously upload new flight plans and other control commands from ground based planning algorithms. Real-time aircraft telemetry is utilized in ground based processing, including GPS position and velocity (±2m, ±0.1m/s respectively) air data, attitude estimates, static wind estimates and other control data. The attitude solutions are real time estimates obtained using measurements from Cloud Cap's Crista IMU, which provide high bandwidth, angle-rates and accelerations ($\pm 300°$/s at 16 bits and $\pm 10g$ at 16 bits respectively). This information is obtained at a rate of 1Hz through the robust 912MHz link providing sufficient bandwidth for high level commands.

Benefits of purchasing a commercially available system are the significant time and effort saved in developing the required infrastructure to tune and test the system. The well-designed and user-configurable Cloud Cap architecture comes complete with a high fidelity hardware-in-the-loop (HWIL) simulation mode that enables real-time testing of the system on the ground before flight tests are performed. While this is primarily meant to simulate the system for controller tuning purposes, it is also interfaced with the planning system so that multi-vehicle simulations can be executed with high levels of accuracy on the ground.

Figure 8 displays the setup of the system with the avionics performing HWIL simulations and the planning system in the loop. The groundstation communicates with each of the avionics through the 912MHz data

[a] Similar projects at MIT had taken 4 years to develop their own autopilot. Recent projects at Stanford University have taken similar periods of time.

link, and the telemetry data from the vehicles is passed on to the planning system through integrated TCP/IP protocols, exactly as would be performed during an actual flight test. An integrated GUI is connected to the system to allow user feedback to the planning system and visualize the state of the mission in real-time using FlightGearV0.9.2 operated in network connection mode. While performing HWIL simulations, each of the avionics is connected through a USB-CAN adapter to a simulator CPU which stimulates the avionics sensors with simulated measurements. The HWIL simulator application allows the specification of a detailed aircraft model that is built up from aircraft geometry and inertia measurements or alternatively specified through calibrated wind tunnel data. By specifying the appropriate aircraft parameters and selecting suitable models for sensors onboard, actuator delays and turbulence parameters, an accurate simulation of flying characteristics can be built up. This is an essential part of the process in validating control settings and testing the performance of the system before attempting an actual flight.

C. Payload Sensors

Additional video and magnetometer sensors have also been integrated onboard the aircraft to provide added real-time measurements about the environment. The pan/tilt video camera transmits video over the 2.4GHz band to the ground-station where it can be processed to track ground objects. Figure 9 shows a captured image from the video system in flight. The onboard magnetometer provides true heading estimates of the aircraft in flight, which can also be used to provide estimates of the ambient winds acting on the vehicle.

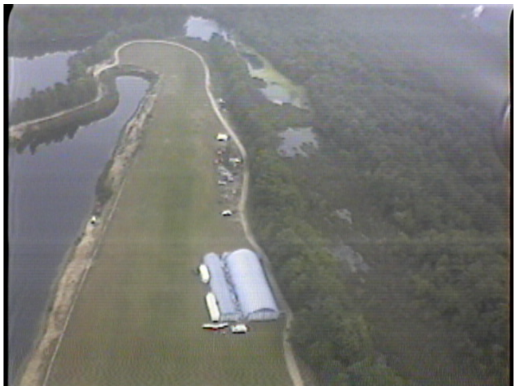

Figure 9. Overhead video of the local flying field at Crow Island in Maynard, MA taken with the onboard video system.

IV. Examples

A. Receding Horizon Control

The UAV testbed has been operated autonomously on numerous occasions. Figures 10–13 shows the results from a mission flown on the UAV testbed using receding horizon control (RHC) to generate waypoint plans in real time. In this scenario, the goal locations used in the planner (shown as circles) were set in a 400m × 600m box pattern over the extents of the flying field and *timing constraints*[3] were enforced to ensure a clockwise sequencing. Figures 10–13 show the progression of the optimal planned paths and the telemetry data from the vehicle during one circuit. Each "×" segment represents one waypoint plan which is returned from the optimal trajectory designer and is uploaded piecewise to the UAV as the MILP optimization completes. The plans are spaced at 100 meters or approximately 5 seconds from the next in the sequence. With 2.4GHz Dell laptops running CPLEXv8.0, these computations typically take less than 1 second, providing substantial margin in completion times.

The gray regions in Figure 14 represent obstacle locations which were encoded into the scenario using mixed integer constraints[13] to constrain the planned trajectories within a safe operating distance. Although no dynamic changes were made to the environment in this scenario (i.e. pop-up obstacle, goal discovery), the framework of the RH controller also allows for this capability.

This flight test was conducted in the presence of winds approximately 25% of the vehicle airspeed, and in a confined area, requiring minimum turn radius constraints to be enforced. As a result of these conditions, the low level vehicle controller is saturating at the maximum bank angle, causing roughly 40 meter overshoot offsets to be flown in some instances. Although the low level autopilot controllers were subsequently tuned to obtain better performance, this flight test highlighted the need for feedback on the planning level to account for the wind estimation error present.

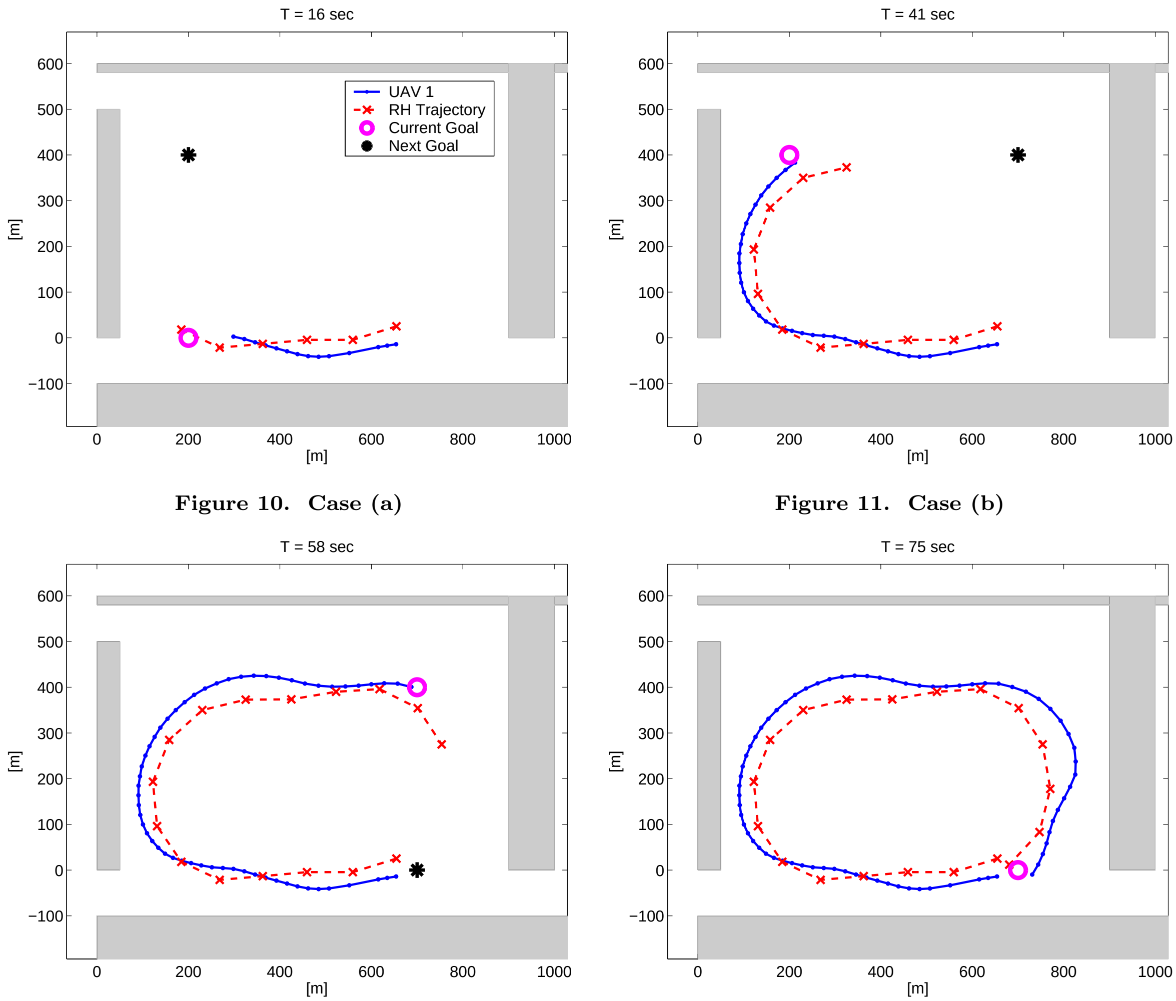

Figure 10. Case (a)

Figure 11. Case (b)

Figure 12. Case (c)

Figure 13. Case (d)

Figure 14. Demonstration of UAV receding horizon control – planned paths are the dashed lines, vehicle telemetry shown by solid lines. Wind disturbances are ~ 5 m/s (W to E).

B. Two Vehicle Formation Flight with Autonomous Rendezvous Using Timing Control

Figure 15 shows the results of an 22 minute autonomous flight involving two UAVs simultaneously flying the same flight plan. Both vehicles tracked the waypoints in the presence of wind, and open loop formation flight was achieved by adjusting the commanded speed until the vehicles were in phase with one another. A 50 meter altitude offset was applied to one of the vehicle trajectories in Figure 15 to allow for easier viewing.

As another practical application for timing control, two UAVs were linked to the same receding horizon trajectory planner and independent timing control was performed about the designed plans. An altitude offset of 20 meters was applied to the second vehicle in order to avoid collisions. Again, both vehicles tracked the waypoints in the presence of wind, and formation flight was achieved through autonomous control of the reference airspeed. Figure 16 shows an aerial photo from the onboard camera as the second UAV autonomously overtook the leader and then slowed down to the desired speed.

Figure 17 shows the relative position error for the two UAVs after the LQG timing controller (see Figure 2) was enabled. The position error is shown to converge to within a 25m × 25m box around the origin, which corresponds to a timing error of ±1 seconds as regulated by the LQG timing controller. Wind disturbances during this flight test were on the order of 1m/s, which corresponds to a magnitude ratio of roughly 5%. The relative position errors shown in the plot show the vehicles maintaining coordinated flight despite the moderate disturbance levels acting on the system.

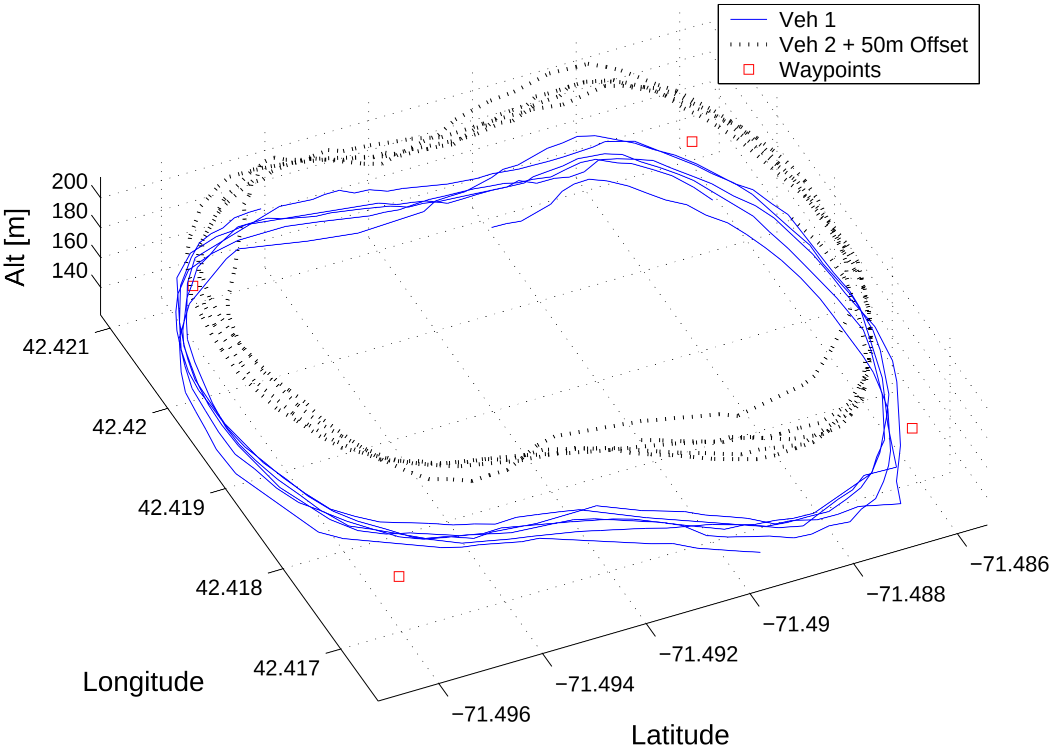

Figure 15. Autonomous UAV flight data. Each vehicle flew the same waypoint plan. The results are shown with a 50m offset for easier viewing.

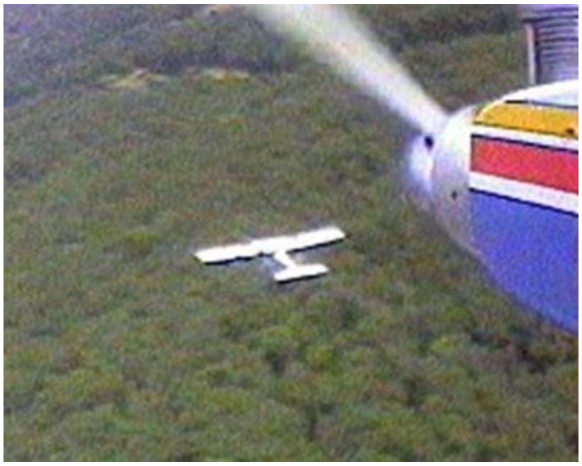

Figure 16. Aerial photo from the onboard camera during the autonomous rendezvous of two aircraft using timing control

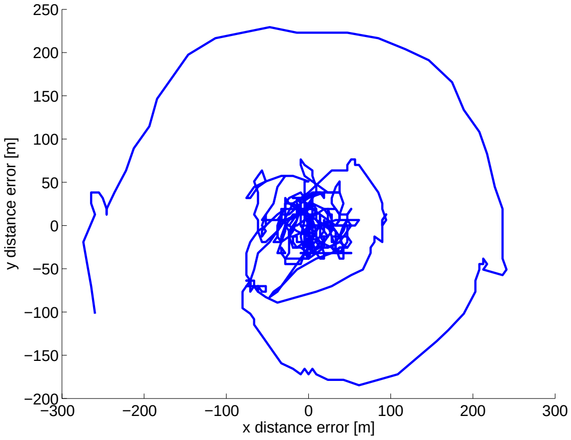

Figure 17. Flight data showing 2 vehicle synchronization and rendezvous.

V. Conclusions

This paper presented hardware demonstrations of the receding horizon task assignment and trajectory design on a new UAV testbed. This multi-vehicle testbed provides a unique platform to evaluate various distributed coordination and control strategies. Future work will integrate distributed collision avoidance formulations, task assignment with the formation of dynamic sub-teams, and missions with heterogeneous vehicles.

Acknowledgments

Research funded by AFOSR Grant #F49620-01-1-0453 and DURIP Grant #F49620-02-1-0216.

References

[1] O. of the Secretary of Defense, "http://www.acq.osd.mil/usd/uav_roadmap.pdf," Tech. Rep., December 2002.

[2] P. R. Chandler, M. Pachter, D. Swaroop, J. M. Fowler, J. K. Howlett, S. Rasmussen, C. Schumacher, and K. Nygard, "Complexity in UAV Cooperative Control," in *Proceedings of the American Control Conference*, Anchorage AK, May 2002.

[3] M. Alighanbari, Y. Kuwata, and J. P. How, "Coordination and Control of Multiple UAVs with Timing Constraints and Loitering," in *Proceedings of the American Control Conference*. IEEE, June 2003.

[4] A. Richards, Y. Kuwata, and J. P. How, "Performance Evaluation Of Rendezvous Using Model Predictive Control," in *AIAA Guidance, Navigation, and Control Conference*. AIAA, August 2003.

[5] J. Bellingham, M. Tillerson, A. Richards, and J. How, "Multi-Task Allocation and Path Planning for Cooperating UAVs," in *Second Annual Conference on Cooperative Control and Optimization*, Nov 2001.

[6] A. Richards, J. Bellingham, M. Tillerson, and J. How, "Coordination and Control of Multiple UAVs," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Monterey, CA, Aug 2002.

[7] G. Laporte and F. Semet, "Classical heuristics for the capacitated VRP," in *The Vehicle Routing Problem*, P. Toth and D. Vigo, Eds. Philadelphia: SIAM, 2002.

[8] L. Bertuccelli, M. Alighanbari, and J. How, "Robust planning for coupled, cooperative UAV missions," *to appear at IEEE the Conference on Decision and Control*, 2004.

[9] M. Alighanbari, "Task assignment algorithms for teams of UAVs in dynamic environments," Master's thesis, MIT, June 2004.

[10] E. King, Y. Kuwata, M. Alighanbari, and J. How, "Coordination and control experiments for UAV teams," *American Aeronautical Society*, 2004.

[11] J. Bellingham, A. Richards, and J. How, "Receding Horizon Control of Autonomous Aerial Vehicles," in *Proceedings of the American Control Conference*, May 2002.

[12] Y. Kuwata and J. How, "Stable Trajectory Design for Highly Constrained Environments using Receding Horizon Control," in *Proceedings of the American Control Conference*. IEEE, June 2004.

[13] A. Richards and J. P. How, "Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming," in *Proceedings of the American Control Conference*, Anchorage, AK, May 2002.