International Journal of Systems Science
Vol. 37, No. 6, 15 May 2006, 385–398

Taylor & Francis
Taylor & Francis Group

Experimental demonstration of coordinated control for multi-vehicle teams

ELLIS KING*, YOSHIAKI KUWATA and JONATHAN P. HOW

Department of Aeronautics and Astronautics, Aerospace Controls Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

(*Received November 2004*)

This paper discusses several unique hardware testbeds that were developed to evaluate coordination and control algorithms for missions with multiple unmanned aerial vehicles (UAVs). The first testbed uses eight rovers and two blimps operated indoors to emulate a heterogeneous fleet of aircraft. The second testbed has eight UAVs flown autonomously with commercial autopilots, which are also used to perform sophisticated hardware-in-the-loop simulations. Future missions will require that teams of UAVs cooperate in dynamic and uncertain environments, and these testbeds can be used to compare various control approaches. A hierarchical configuration of task assignment, trajectory design, and low-level waypoint following uses a receding horizon framework to control the team of vehicles. The resulting demonstrations of the high-level planning algorithms on scaled vehicles operating in uncertain environments represent key steps towards transitioning them to future UAV missions.

Keywords: UAV; Task assignment; Trajectory design; Coordinated control; Multi-vehicle control

1. Introduction

Autonomous Unmanned Aerial Vehicles (UAVs) offer several advantages over conventional manned aircraft because they can be used in situations otherwise too dangerous for pilots, and, because they save the weight of the pilot support systems, UAVs can stay aloft on very long surveillance missions (OSD 2002). While the capabilities of UAVs are growing, current control structures were conceived with limited roles in mind for these vehicles. Thus, in order to fully exploit these expanding capabilities, it is necessary to extend the existing control and planning technologies. This requires developing techniques to optimize the coordination of multi-vehicle teams, which is comprised of the coupled subproblems of determining sub-team composition, allocating resources (task assignment), and optimizing vehicle trajectories (Chandler *et al*. 2002). 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 (Samad and Balas 2003, Richards *et al*. 2003, Alighanbai *et al*. 2003, Butenko *et al*. 2003, Kott 2004, Grundel *et al*. 2004), but a key step towards transitioning these high-level algorithms to future missions is to successfully demonstrate that they can handle similar implementation challenges using scaled vehicles operating in realistic environments. Performing experiments on scaled testbeds will highlight the fundamental challenges associated with (i) planning for a large team in real time with computation and communication limits; (ii) developing controllers that are robust to uncertainty in situational awareness, but are sufficiently flexible to respond to important changes; and (iii) using communication networks and distributed processing to develop integrated and cooperative plans.

Recent tests (Valenti *et al*. 2004, Schouwenaars *et al*. 2006) have demonstrated high-level planning algorithms with two real aircraft (both manned, one acting as a UAV flying a support mission), but this

*Corresponding author. Email: etking@alum.mit.edu

International Journal of Systems Science
ISSN 0020–7721 print/ISSN 1464–5319 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207720600798320

paper presents three testbeds that have been developed at MIT to perform more extensive tests with larger teams of unmanned vehicles. One uses multiple rovers and blimps operated indoors to emulate a heterogeneous fleet of vehicles performing a search and rescue mission. The second uses two distinct classes of aircraft to form teams of UAVs that fly autonomously using a commercial autopilot available from Cloud Cap Technology. The third uses the same autopilots in a detailed hardware-in-the-loop (HWIL) simulation. Note that the design choice of using a commercial autopilot on 'almost ready to fly' (ARF) aircraft limits the degree of possible customization, but recent flight tests have clearly shown the advantages of this approach, such as the ease of achieving operational status, the tolerance of the equipment to mishaps, and the ability to perform sophisticated missions. The paper discusses the limitations, benefits, and roles of these testbeds and shows that the combination of all three provides a unique facility for analyzing all aspects of the coordination and control algorithms for future multi-UAV missions.

2. Coordination algorithms

This section discusses guidance and control algorithms for a fleet of cooperating UAVs. Figure 1 shows the overall control system, which includes the goal assignment, resource allocation, and trajectory optimization problems. For many vehicles, obstacles, and targets, fleet coordination is a very complicated non-convex optimization problem. 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 (Bellingham *et al*. 2001, Richards *et al*. 2002). The decomposition approach simplifies the coupling between the assignment and trajectory design problems by calculating and communicating only the key information that connects them. Our approach uses an approximate cost-to-go calculation to obtain good estimates of the costs associated with feasible paths around obstacles in the environment, such as buildings and no-fly zones. These costs are included in the assignment problem that is solved using the *petal* algorithm (Bellingham *et al*. 2001, Laporte and Semet 2002). Uncertainty in the target classification due to poor or conflicting information enters the problem as uncertainty in these assignment costs. As demonstrated by Bertucccelli *et al*. (2004), 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.

2.1 *Task assignment*

While the decomposition approach greatly reduces complexity, the problem of task assignment with precedence constraints is NP-Hard, and achieving the exact solution for a large team of UAVs is computationally intensive and not suitable for real-time applications. Our implementation of the petal algorithm uses a heuristic method to prune out the solutions that are not likely to be a part of the optimal solution, which significantly

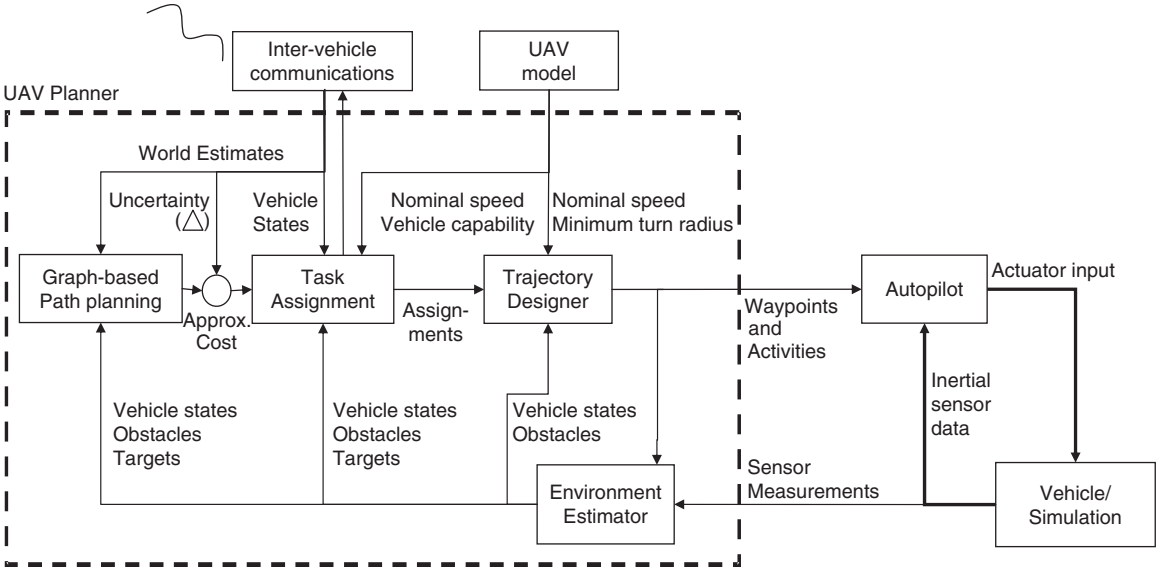

Figure 1. System algorithm architecture for the UAV testbed.

speeds up the task assignment process. While this approach has been shown to work well for small problems (Richards *et al*. 2002), it is still difficult to run in real time for problems with many vehicles and tasks. To perform reassignment in real time, as required in a frequently changing environment, we have extended the petal method to develop a receding horizon task assignment (RHTA) algorithm (Alighanbari 2004).

The RHTA significantly reduces the computation time by selecting at most m (typically $m < 3$) tasks for each UAV and then repeats the optimization over several iterations. The process is to add the first of the selected tasks to the mission list for each UAV, update each UAV position and time, remove the assigned tasks from the task list, and then repeat until all tasks are assigned. Selecting multiple tasks (i.e., $m > 1$) during each iteration provides better insight into the implications of each assignment and enables much greater coordination between the UAVs in the team. As with any receding horizon controller, the performance of the algorithm strongly depends on the cost-to-go function. Our RHTA uses a unique approach that includes constraints to ensure that sufficient munitions exist to complete the rest of the mission after the team finishes the assigned tasks. These constraints in effect give a low-detail future plan, which helps eliminate a key failure with iterative greedy-type algorithms (Schumacher *et al*. 2001). Timing and precedence constraints can be imposed using the approach by Alighanbari *et al*. (2003) or by simply requiring that tasks be removed from the list until all precedents have been assigned. The first of these two is more complicated, but is typically less conservative than the second.

2.2 *Trajectory optimization*

The final step is to design detailed UAV trajectories around the obstacles. Our controller optimizes the paths using a MILP-based receding horizon planner (Bellingham *et al*. 2002), which has been proven to guarantee the arrival of the UAVs at the target in bounded time (Bellingham *et al*. 2003). The RH-MILP (or RH-TRAJ) technique 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 (Kuwata and How 2004). Novel pruning and graph search algorithms have also been included, and these significantly reduce the computational load. The trajectories shown in the following experimental sections are designed in real time using RH-MILP.

2.3 *Control architecture*

Figure 1 shows the control architecture used for the UAVs, but the setup is very similar for the rover/blimp testbed (Richards *et al*. 2003). The low-level control and basic estimation tasks are run onboard, but 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 planner updates the cost map, re-assigns the tasks, and/or re-designs the trajectories.

2.4 *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 assigned tasks and designed trajectories are feasible. As shown in figure 2, the aircraft is influenced by wind disturbances that consist of averaged, static components ($\overline{W}$) and turbulence (δW), which is captured using a Dryden model. These winds limit the aircraft's ability to follow the desired flight plan and execute the tasks at the desired times. Three components of the control system in figure 2 are used to compensate for these errors.

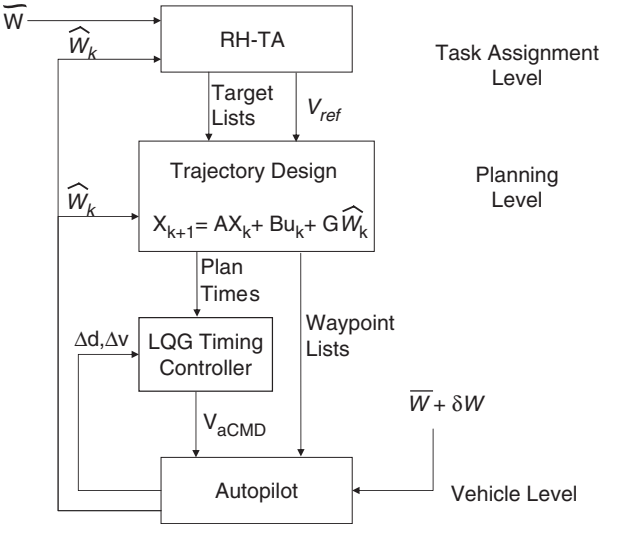

Figure 2. Multi-level estimation and timing planning/control approach used to compensate for environmental disturbances.

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. A key point 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, the trajectory design and task assignment levels must compensate for the errors, but these two high-level planners must also be robust to the uncertainty in the cost calculations introduced by errors in the wind estimates. The trajectory optimization uses 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 receding horizon task assignment algorithm (RRHTA) uses estimates of the static wind $\hat{W}_k$ and wind uncertainty $\tilde{W}$ to choose assignments that are robust to the possible variations in the vehicle flight times. This algorithm also chooses the reference velocity V_{ref} required to enforce the loitering constraints within the vehicle minimum and maximum saturation limits. Note that wind is only one of the many ways that uncertainty enters this system, and Bertuccelli *et al.* (2004) and Tin (2004) discuss techniques to account for these other effects.

3. Testbed infrastructure

The testbeds have been designed to simulate many challenging operational scenarios, with a particular focus on cooperative coordination and control of multiple vehicles for missions such as: low-cost multi-target surveillance and tracking, wide-spread search, and moving target location and tracking. These missions typically require the close coordination and control of many different types of vehicles (e.g., unmanned vehicles {fighters, strike, and electronic suppression}, semi-autonomous UAVs {fixed-wing and helicopters}, surveillance aircraft and satellites, communication vehicles {Airborne Warning and Control System – AWACS}, and ground forces) to accomplish the overall objectives. The testbeds were designed to reflect the complexity expected in future combat operations and consist of many (semi-) autonomous heterogeneous vehicles. We used *simple* vehicles, such as ARF UAVs, blimps, and rovers, so that *many* of them can operate together. This provides a good combination of flexibility, agility, and mobility, and allows us to use the testbeds in a broad range of applications.

The system infrastructure was set up to emulate a fully integrated fleet of UAVs, but a key part of the design philosophy was to maintain as much simplicity as possible in the vehicles themselves, reducing the conservatism that tends to exist for more expensive UAV platforms. As such, all high-level processing is executed off-board using the planning laptops and all data passes through a central hub that performs data management between the planning computers and vehicles. This central hub is used to simulate delays and outages of the communication between vehicles, emulate the payload sensor measurements, and detect changes in the environment. Data and commands can be transferred between the planning and vehicle systems at a rate of up to 1 Hz, providing a fast response to dynamic changes detected in the environment.

While this setup greatly reduces the logistics required to operate a testbed, it still provides the functionality needed to evaluate high-level planning algorithms. Figure 3 illustrates how the testbeds 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. Since the robustness and performance of these links far exceed the capabilities of actual intra-vehicle communications, the connections can be neglected and we can focus on the communication models used in the simulation. Current work is evaluating the performance and computation trade-offs for various control architectures in the task assignment and path planning optimizations.

Table 1 compares the three testbeds in terms of the types of experiments that can be performed, the uncertainty included in these experiments, and the limitations that exist. For example, while tests on the rovers/blimps have limited realism, they are a versatile platform for carrying new sensors and they provide an easy way to investigate the performance of new control algorithms with realistic limits on the computation and communication. The HWIL testbed was primarily designed to test the autopilot settings before flights; however, the high fidelity HWIL simulation can also be used to perform detailed experiments of multi-vehicle flights that would otherwise not be possible on the vehicle testbed due to logistical constraints. The HWIL results are realistic because the vehicle and environment models in this simulation were calibrated using experimental flight data, and the planning system interacts with the autopilots exactly as it would if the aircraft were actually flying. Of course, the UAVs provide the final hardware validation, but due to logistical constraints, the scenarios tend to be quite simple and only one or two UAVs are used at a time. Maintaining three testbeds is clearly difficult, but, as indicated in table 1, each plays an important role in evaluating all the aspects of the coordination and control problem (computation, communication, vehicle dynamics, and uncertainty). Furthermore, because the interfaces between the planning system and the vehicles were designed to be identical for all testbeds, it is very easy to transition the

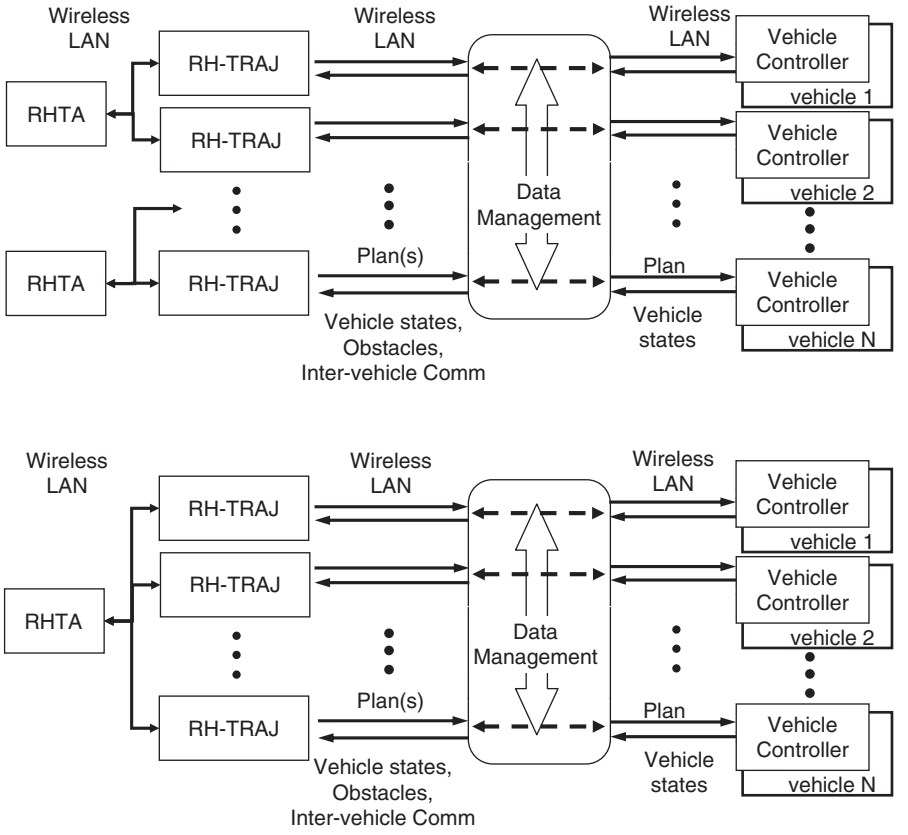

Figure 3. Control architecture choices. Top: decentralized path planning/distributed task assignment; bottom: decentralized path planning/centralized task assignment.

Table 1. Testbed comparison.

	Rovers & blimps	Autopilot HWIL	UAVs
Experiment uncertainty	Scenario outcome Communication Computation	Dynamics Disturbances Communication Computation	Dynamics Disturbances Communication Computation
Utility	Versatile platform Heterogeneous Complex scenarios # vehicles ($N \geq 8$)	Full vehicle dynamics Preflight validation Complex scenarios # vehicles ($N \approx 8$)	Realistic experiments Sensor platform Hardware validation Three-dimensional
Limitations	Two-dimensional Few disturbances Simplified dynamics Moderate logistics	A simulation	Heavy logistics Simple scenarios Vehicle numbers ($N \leq 3$) and scalability

control algorithms from one testbed to another, which significantly reduces the logistical problems.

3.1 *Rover/blimp testbed*

The first testbed uses multiple rovers and blimps operated indoors to emulate a heterogeneous fleet of vehicles that could be used to perform Suppression of Enemy Air Defense (SEAD) type missions. The rovers in figure 4 are ActivMedia's P3-ATs, which are operated with minimum speed and turn rate constraints to emulate the motion of an aircraft. A Sony VAIO mounted on the rover processes sensor data and performs the low-level control, while all high-level planning is done off-board

using 2.4 GHz Dell laptops running MATLAB and CPLEX. A direct wireless Ethernet connection provides a fast and reliable network between the laptops, so this is essentially the same as having both laptops onboard. The ArcSecond Constellation 3D-i is used to measure the vehicle position indoors. This sensor uses laser metrology to provide ±4 mm position accuracy at 20 Hz. The rovers can also be outfitted with the Canadian Marconi Company Superstar^{TM} receiver for outdoor operations. The video cameras on the rovers are Axis EVI-D30 with remote controlled pan/tilt and object tracking capabilities. Each camera outputs analog video to an Axis 2401 video server which converts the signal into digital video and transmits it over Ethernet to a wireless bridge. This digital video is then transmitted to the ground station to display the results for the operator of the team. The video cameras also provide measurements of the relative bearing angle to the target, which are used in an extended Kalman filter to provide real-time estimates of the target locations for the task assignment algorithms. This work highlighted the need to include robustness to the tracking and estimation errors in the task assignment (Bertuccelli *et al.* 2004).

The 7 ft diameter blimps used in the testbed provide a simple means to include vehicles with different dynamics and capabilities. The blimps were designed to perform tasks in conjunction with the rovers, such as rapidly performing reconnaissance or classification beyond the obstacles blocking the rovers. They were scaled to carry the VAIO and have an identical control architecture. While these vehicles are more complicated to operate than the rovers, they are far easier to fly indoors than fixed or rotary wing vehicles.

Figure 4. Four of eight ActivMedia P3-AT rovers.

3.2 *Tower Trainer 60 UAV*

Figure 5 shows the second testbed, which started as a fleet of eight UAVs. In order to make successful demonstrations of multi-vehicle flights, the logistics require that all the vehicles 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 min 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 the additional weight of the sensors and batteries. The vehicles selected for the testbed are commercially available Tower Trainer 60 aircraft, which have easy handling characteristics and relatively large payload capacities. Only minor modifications are required to augment these ARF aircraft to suit the mission requirements, so they can be quickly constructed and standardized across the entire fleet. In addition, the use of cheap standardized aircraft for the fleet simplifies maintenance, repairs, and flight logistics. The Trainer 60 airframe has proven to provide a very stable and robust platform for the types of flight experiments detailed in later sections.

A single ARF 60 aircraft is shown in figure 6, with a list of important aircraft parameters in table 2. The large wing area of the aircraft, combined with the four-stroke, Saito-91s (91 ccs) engine provides more than 3 lbs of

Figure 5. Fleet of eight identical Trainer 60 aircraft used in the multi-UAV testbed at MIT.

Table 2. Airframe comparison.

	Monocoupe	Tower Trainer 60
Wing span (in.)	96.5	69.5
Wing area (in.^{2})	1460	877
Dry weight (lbs)	18	10.5
Engine power (hp)	4	1.5
BFC (oz/min)	1	1.2
Fuel capacity (fl-oz)	64	48
Flight duration (min)	120	40
Payload capacity (lbs)	5	2
Payload volume (in.^{3})	$9 \times 9 \times 9$	≈ 0
Flaps	Yes	No
Flight envelope (mph)	40–68	50–60

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 enables flights longer than 40 min 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 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, this inherent stability also limits the aircraft maneuverability. The reduction in performance, combined with minimum flight speeds of approximately 20 m/s, requires that slightly larger test areas be utilized to perform effective demonstrations. However, this is a good trade-off for the proof-of-concept missions that are the focus of this phase of the project.

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 2.4 GHz band to the ground-station where it can be processed to track ground objects. 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.

3.3 *Autopilot*

At the time the aircraft testbed was designed, the options for low-level vehicle control included either purchasing or constructing autopilots. With proprietary source code restrictions, purchasing an autopilot has less flexibility, but the resulting reduction in the time to achieve operational status is significant. Experience has shown that similar projects at MIT and other universities have typically taken 3–4 years to develop their own autopilots. As a result, we decided to use a commercial autopilot, and after considering various options, the Piccolo^{TM} autopilot from Cloud Cap Technology was selected (Vaglienti *et al*. 2004). This autopilot is used onboard the aircraft to perform the autonomous vehicle stabilization and waypoint navigation.

Communication between the ground station and the aircraft is done using a 1 W transmitter over the 912 MHz datalink, which permits the vehicles to navigate up to 3 miles from the base. New flight plans and other control commands from ground based planning algorithms can be uploaded using this communication link. Real-time aircraft telemetry, including GPS position and velocity (± 2 m, ± 0.1 m/s respectively), air data, altitude estimates, and static wind estimates are down-linked and used in the displays and control algorithms on the ground. Cloud Cap's Crista IMU provides high bandwidth angle-rates and accelerations ($\pm 300^\circ$/s at 16 bits and ± 10 g at 16 bits respectively), which are used to compute real-time estimates of the aircraft altitude. All down-linked data is obtained at a 1 Hz rate, which provides sufficient bandwidth for the high-level commands.

The 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 also has a high fidelity HWIL simulation mode that enables real-time testing of the system on the ground before flight tests are performed. The primary purpose of the HWIL capability is to simulate the system for controller tuning, but we also developed an interface to our planning system so that multi-vehicle simulations can be executed with high levels of accuracy on the ground. These combined benefits greatly offset the issue of the autopilot

customization being constrained to only changes in the control loop gains.

Figure 7 displays the setup of the system with the avionics performing HWIL simulations and the planning system in the loop. The ground station communicates with each of the avionics through the 912 MHz data link, and the telemetry data from the vehicles is passed to the planning system through the Ethernet link exactly as in an actual flight test. The system also has an integrated GUI that enables user feedback to the planning system and displays the state of the mission in real-time using FlightGear V0.9.2 operated in network connection mode. When 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 the onboard sensors, actuator

Figure 6. Comparison of the Trainer 60 and Monocoupe aircraft in the MIT UAV testbed.

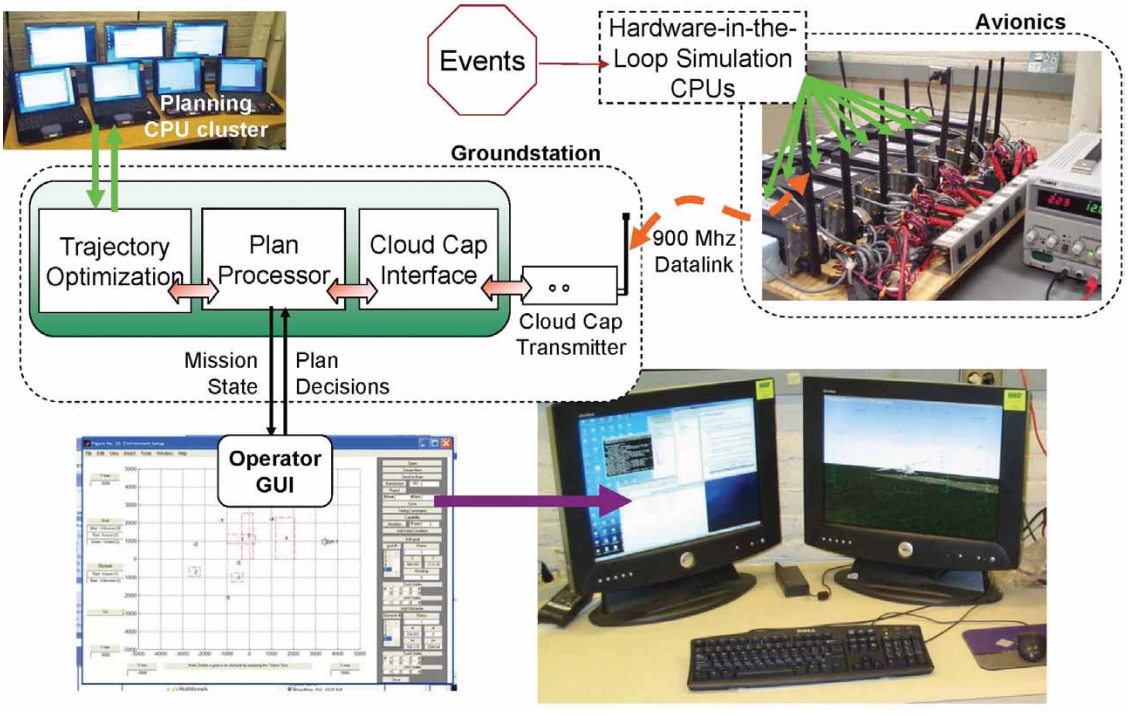

Figure 7. HWIL configuration allows simultaneous simulation of eight aircraft with the integrated planning system connected through the 912 MHz data link, exactly as is performed in flight.

delays, and turbulence parameters, it can accurately simulate flight characteristics. This calibration step is an essential part of the process in validating control settings and testing the performance of the system before attempting an actual flight.

3.4 *Monocoupe UAV*

A second type of UAV has been added to the testbed using a Monocoupe airframe, which has several advantages over the Trainer 60. As shown in table 2, the Monocoupe has almost double the overall wing area, which allows for a payload capacity of up to 5 lbs (not including fuel and onboard avionics). These vehicles can also be purchased as almost ready to fly, so they retain the operational simplicity originally envisaged for the UAV testbed. The current payload is a 3.5 lb high resolution pan/tilt camera system (same Axis EVI-D30 used on the rovers) positioned in the fuselage belly. The camera is controlled directly through the avionics and can automatically track high contrast objects during flight. Further payloads can either be slung underneath the belly of the aircraft or held inside the 9 in. square cabin area. The current testbed has four Monocoupes that fly with the tower trainers in a heterogeneous team.

The Monocoupe can achieve a two hour flight duration through the use of a gasoline-powered 2-stroke engine in place of the nitro-methane powered 4-stroke engines used on the trainers. These engines have almost 3 times the displacement of the trainer engines, yet have a lower fuel consumption rate. The fuel capacity is currently 64 oz, but if there is no additional payload, the Monocoupes could carry additional fuel tanks to increase the overall fuel capacity to 164 oz, enabling flight times of nearly four hours. Since the payload compartment is directly on the center of gravity, fuel loss during flight does not cause any adverse balance effects. The net result is that this UAV offers extended range, duration, and payload capacity.

Large control surfaces coupled with high-torque actuators greatly improve the overall performance of the autopilot control loops for the Monocoupe. When calibrated properly, the Monocoupe exhibits very robust flight path tracking while under autopilot control, even in high wind conditions. The effect is to allow greater flexibility and capabilities for the UAV testbed, but just as importantly, completing the aircraft construction and transitioning the autopilot was done in less than 3 weeks.

4. Experimental results

4.1 *Experiments using the rover/blimp testbed*

Figure 8 shows an experimental result on the rover/blimp heterogenous testbed. The scenario models the environment where the information is partially

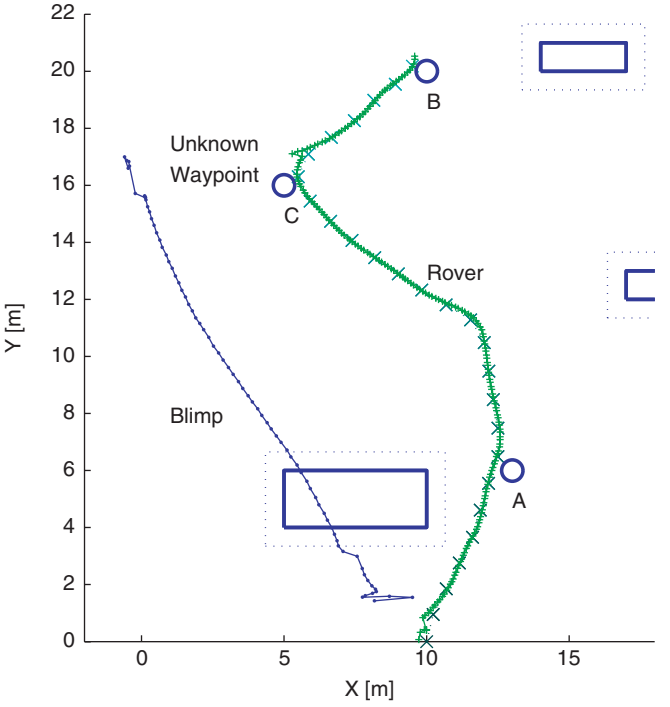

Figure 8. Rover and blimp trajectories recorded during a heterogeneous reconnaissance and strike demonstration.

available, requiring search missions to be executed concurrently with the strike. The obstacles impede the rovers, but the blimp can fly over them. In this experiment, the blimp searches for targets while the rover executes tasks. The trajectory on the left shows the path of the blimp performing its search pattern, and the trajectory on the right shows the path of the rover as it navigates to each task. For the purpose of this test, a heading command is sent to the blimp, while the rover follows the waypoints (marked with $\times$) generated by the RH-TRAJ. Target/task locations are marked with $\bigcirc$. Initially the target C was not known, but the blimp is sent to perform a reconnaissance of the open space to the left, flying over the obstacle at the bottom of the figure. While the rover is enroute to execute task B, the blimp discovers the new task (waypoint C) and RHTA reassigns that task to the rover. This initial result demonstrates the successful integration of the heterogenous vehicles in our planning system and future tests will incorporate robustness into the task assignment for a heterogeneous system of rovers and blimps in an uncertain environment.

4.2 *Five UAV HWIL simulation with dynamic tasking*

Figure 9 shows experimental results from one demonstration involving five UAVs and a mixture of both high and low value targets in a complex environment. For this scenario, high value targets (HVT) B, C, and G require both a strike and a subsequent bomb damage assessment (BDA), while the remaining low value targets (LVT) require only a strike task by a

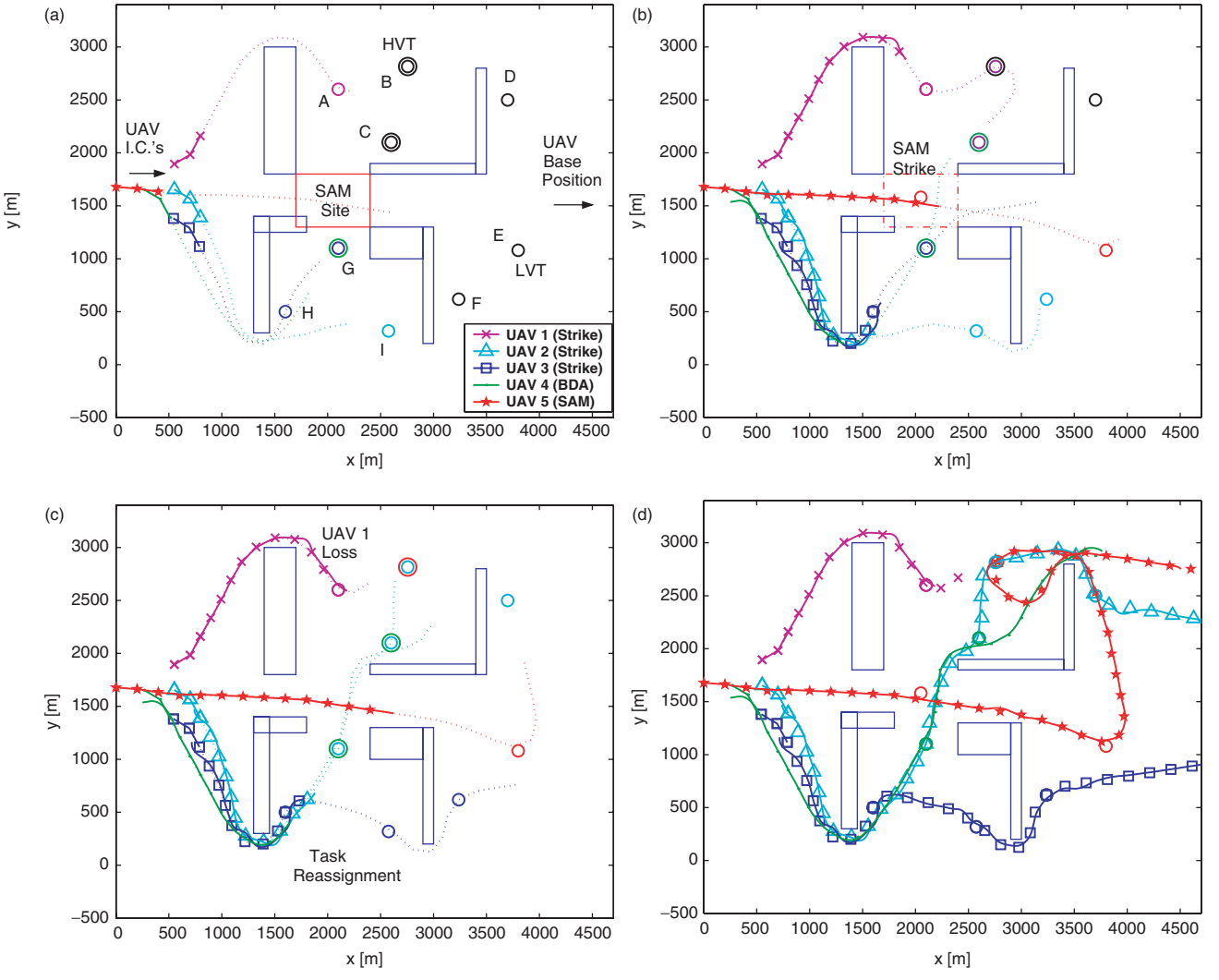

Figure 9. Five UAV mission with dynamic task assignment using RHTA. Paths show both the optimal planned trajectories and the HWIL autopilot responses with simulated wind turbulence. (a) Initial conditions and task assignments. Solid blue lines are obstacles; (b) UAV5 strikes SAM, allows other UAVs to pass; (c) UAV1 lost, requires task re-assignment (d) Completed mission with all tasks achieved. (Color version available online.)

single UAV. UAV 5 is also given the capability to remove the centrally located SAM site, which the other UAVs are not permitted to enter. Initial assignments send strike UAVs 1, 2, and 3 around the obstacles to the targets A, I, and H respectively, while UAV 4 is assigned the first BDA task G (figure 9(a)). Note that a typical mission *timing constraint* is also shown by requiring that strike task D be completed only after the BDA for target B has been accomplished. UAV 5 is assigned to take out the SAM site which would then permit the strike vehicles to pass through the central region (figure 9(b)). In this scenario, UAV 1 suddenly fails after reaching target A, and the remaining tasks are re-assigned to the rest of the team using the RHTA (figure 9(c)). Figure 9(d) shows the completed mission after all tasks have been completed. This scenario demonstrated a real-time implementation of the RHTA algorithm in conjunction with RH-TRAJ on the multi-vehicle HWIL testbed. This type of complicated mission would be very difficult to execute with the actual aircraft, but they can easily be carried out on the HWIL testbed with high confidence in the levels of performance demonstrated.

4.3 *Receding horizon trajectory design*

The UAV testbed has been operated autonomously on numerous occasions. Figure 10 shows results from a mission flown on the UAV testbed using receding horizon control to generate waypoint plans in real time. In this scenario, the goal locations used in the planner (shown as ◯) were set in a 400 m × 600 m box pattern over the flying field and timing constraints were enforced to ensure a clockwise sequencing. Figure 10 shows the progression of the optimal planned paths and the telemetry data from the vehicle during one circuit. The '×' segments are the waypoints of the optimized trajectories that were uploaded piecewise to the UAV as the MILP optimization completed. The waypoints in the optimized paths are approximately 100 m apart (flight time of about 4 s). With 2.4 GHz Dell laptops running CPLEX v8.0, the path design computations typically take less than 1 s providing a substantial margin to assimilate the new set of trajectory waypoints and communicate them to the UAVs. The gray regions in the figures represent obstacle locations that were encoded into the scenario using mixed integer constraints (Richards and How 2002) to keep the planned trajectories within a safe operating distance. This flight test was conducted in the presence of winds that were approximately 25% of the vehicle airspeed and in a confined area, which required that minimum turn radius constraints be enforced. As a result of these conditions, the low-level vehicle controller saturated at the maximum bank angle, causing roughly 40 m offsets at some points. Although the low-level autopilot controllers were subsequently tuned to obtain better performance, this flight test clearly indicated the need for feedback on the planning level to account for wind estimation errors, as shown in figure 2. In addition, along-track speed control has been implemented to ensure that timing constraints are met, a key element in achieving coordination in the fleet.

4.4 *Two-vehicle formation flight with autonomous rendezvous using timing control*

Figure 11 shows the results of a 22 min autonomous flight involving two UAVs simultaneously flying

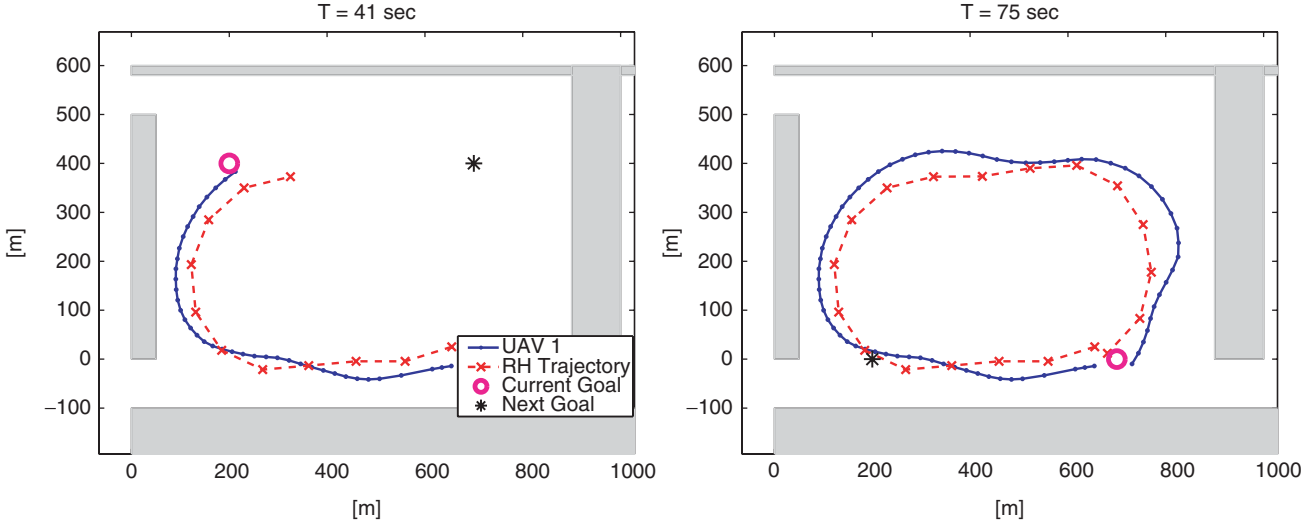

Figure 10. Demonstration of UAV receding horizon control. Planned paths are dashed lines, vehicle telemetry are solid (start at 41 s). Right figure shows completion of a circuit. Wind disturbances ~5 m/s (W to E).

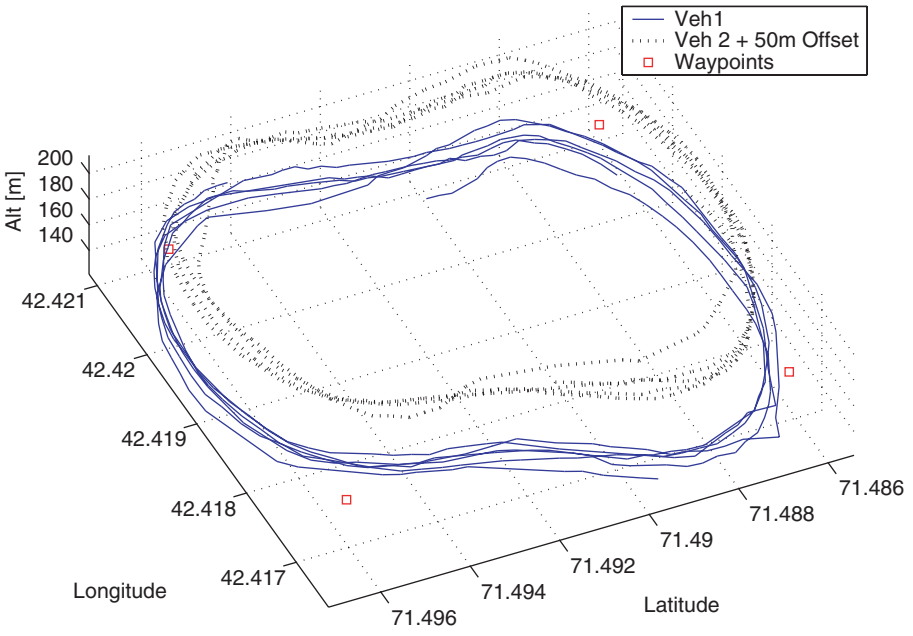

Figure 11. Autonomous UAV flight data. Each vehicle flew the same waypoint plan. The results are shown with a 50 m offset for easier viewing.

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 m altitude offset was applied to one of the vehicle trajectories in figure 11 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 along the designed plans. An altitude offset of 20 m 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 12 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 13 shows the relative position error for the two UAVs after the LQG timing controller in figure 2 was enabled. The position error converges to within 25 m, which corresponds to a timing error of ± 1 s as regulated by the LQG timing controller. Wind disturbances during this flight test were on the order of 1 m/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 12. Aerial photo from the onboard camera during the autonomous rendezvous of two aircraft using timing control.

5. Conclusions

This paper presented a series of testbeds that were developed to demonstrate planning and control algorithms for future missions that will use teams of multiple UAVs. The primary focus was on a UAV testbed that is used to perform the final validation of the algorithms. Choosing to use a commercial autopilot on 'almost ready to fly' aircraft produced a UAV testbed that has a limited degree of customization, but it can perform

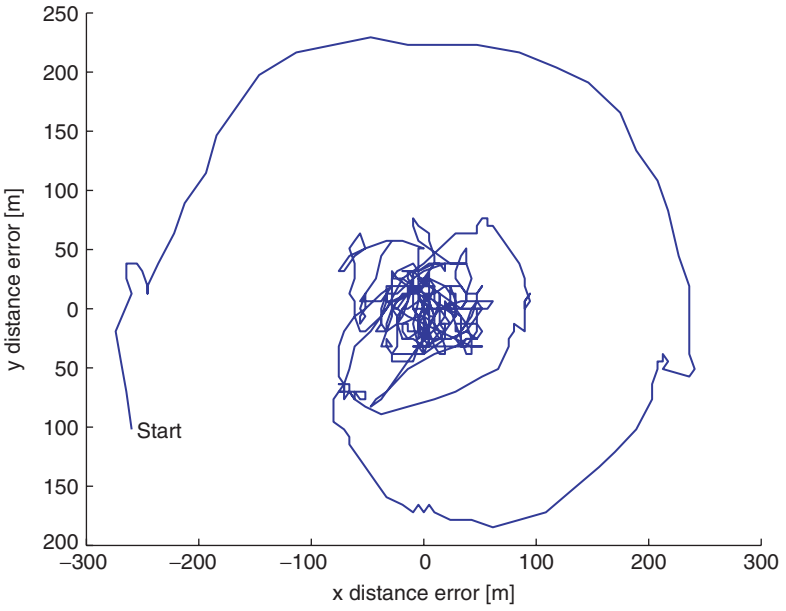

Figure 13. Flight data showing two vehicle synchronization and rendezvous. Plot shows the relative separation error between two UAVs during an autonomous in-flight rendezvous maneuver using LQG timing control.

very sophisticated demonstrations and is very tolerant to mishaps, which is essential when doing numerous flight tests. The local UAV testbed enabled a very aggressive test schedule which experimentally verified the receding horizon task assignment and trajectory design algorithms in the face of external disturbances. However, flight tests with these (and other) UAVs highlighted the need to have other testbeds, such as the rovers and HWIL autopilots, that allow more frequent experiments with fewer logistical issues. The resulting combination of the three multi-vehicle testbeds at MIT provides a unique facility to investigate all aspects of the coordination and control algorithms for future multi-UAV missions.

Acknowledgments

Research funded by AFOSR Grant #F49620-01-1-0453 and DURIP Grant #F49620-02-1-0216. The authors would like to acknowledge the strong support of the large team deployed to complete these experiments, including Mehdi Alighanbari, Luca Bertuccelli, Ian Garcia, Dr. Arthur Richards, Pete Young, Carl Engel, and Dan Harjes.

References

M. Alighanbari, "Task assignment algorithms for teams of UAVs in dynamic environments", Master's thesis. MIT Department of Aeronautics and Astronautics, 2004.

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*, 2003, pp. 5311–5316.

J. Bellingham, A. Richards and J. How, "Receding horizon control of autonomous aerial vehicles", in *Proceedings of the American Control Conference*, 2002, pp. 3741–3746.

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*, 2001, pp. 1–19.

J. Bellingham, Y. Kuwata and J. How, "Stable receding horizon trajectory control for complex environments", in *Proceedings of the AIAA Guidance, Navigation and Control Conference*. AIAA2003-5635, Austin, TX, 2003.

L. Bertuccelli, M. Alighanbari and J. How, "Robust planning for coupled, cooperative UAV missions", in *Proceedings of the IEEE Conference on Decision and Control*, pp. 2917–2923, 2004.

S. Butenko, R. Murphey and P. Pardalos (Eds.), *Recent Developments in Cooperative Control and Optimization*, Vol. 3, Kluwer Academic Publishers, 2003.

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, 2002, pp. 1831–1836.

D. Grundel, R. Murphey and P. Pardalos (Eds.), *Theory and Algorithms for Cooperative Systems*, Vol. 4 of *Series on Computers and Operations Research*, World Scientific Publishing Co, 2004.

A. Kott, *Advanced Technology Concepts for Command and Control*, Xlibris Corporation, 2004.

Y. Kuwata and J. How, "Stable trajectory design for highly constrained environments using receding horizon control", in *Proceedings of the American Control Conference, IEEE*, 2004, pp. 902–907.

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

OSD (2002). Office of the secretary of defense. Available online at: www.acq.osd.mil/usd/uav_roadmap.pdf (Accessed April 2005).

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, 2002, pp. 1936–1941.

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*, AIAA2002–4588, Monterey, CA, 2002.

A. Richards, Y. Kuwata and J.P. How, ''Experimental demonstrations of real-time MILP control, in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, AIAA2003–5802, Austin, TX, 2003.

T. Samad and G. Balas, *Software-Enabled Control: Information Technology for Dynamical Systems*, Wiley-IEEE Press, 2003.

T. Schouwenaars, M. Valenti, E. Feron, J.P. How and E. Roche, ''Linear programming and language processing for human-unmanned aerial-vehicle team mission planning and execution'', *Submitted to the AIAA Journal of Guidance, Control, and Dyanmics*, **29**, 2006, pp. 303–313.

C. Schumacher, P.R. Chandler and S. Rasmussen, ''Task allocation for wide area search munitions via network flow optimization'', in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, AIAA2001-4147, Montreal, Canada, 2001.

C. Tin, ''Robust multi-UAV planning in dynamic and uncertain environments'', Master's thesis, MIT Department of Mechanical Engineering, 2004.

B. Vaglienti, R. Hoag and M. Niculescu, *Piccolo System User's Guide*. Cloud Cap Technology, Available online at (Accessed April 2005): www.cloudcaptech.com/piccolo/piccolo_users_guide.pdf, 2004.

M. Valenti, T. Schouwenaars, Y. Kuwata, E. Feron and J.P. How, ''Implementation of a manned vehicle – UAV mission system'', in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, AIAA2004–5142, Providence, RI, 2004.