

Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery

Mario Valenti*, Brett Bethke† Gaston Fiore‡ and Jonathan P. How§
Massachusetts Institute of Technology, Cambridge, MA 02139

Eric Feron¶
Georgia Institute of Technology, Atlanta, GA 30322

This paper presents flight tests of a unique indoor, multi-vehicle testbed that was developed to study long duration UAV missions in a controlled environment. This testbed uses real hardware to examine research questions related to single and multi-vehicle health management, such as vehicle failures, refueling, and maintenance. The primary goal of the project is to embed health management into the full UAV planning system, thereby leading to improved overall mission performance, even when using simple aircraft that are prone to failures. The testbed has both aerial and ground vehicles that operate autonomously in a large test region and can be used to execute many different mission scenarios. The success of this testbed is largely related to our choice of vehicles, sensors, and the system's command and control architecture, which has resulted in a testbed that is very simple to operate. This paper discusses this testbed infrastructure and presents flight test results from some of our most recent single- and multi-vehicle experiments.

I. Introduction

Unmanned vehicles have been in use for many years. Today, military and intelligence organizations are using Unmanned Aerial Vehicles (UAVs) to locate, assess, and attack targets from sophisticated operator stations miles from the battlefield. However, many unmanned vehicles do not exhibit the level of performance and flexibility needed to complete an entire mission autonomously. For example, most UAV guidance and mission planning systems do not possess the capability to recognize and react to unexpected changes in the operating conditions. The complexity of this problem increases as multiple agents are introduced. For example, if another autonomous agent is

*Student Member AIAA, Ph.D. candidate, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 valenti@mit.edu

†Student Member AIAA, S.M. Candidate, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 bbethke@mit.edu.

‡Student Member AIAA, S.M. Candidate, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 gafiore@mit.edu.

§Associate Fellow AIAA, Associate Professor of Aeronautics and Astronautics, Aerospace Controls Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave, M.I.T. Rm 33-326, Cambridge, MA 02139 jhow@mit.edu Author to whom all correspondence should be addressed

¶Associate Fellow AIAA, Professor of Aeronautics and Astronautics, Georgia Institute of Technology, Atlanta, GA 30322 eric.feron@aerospace.gatech.edu

added to the mission scenario, then both vehicles must resolve information regarding the impending actions of the other vehicle. Similarly, if a manned agent is added to the original scenario, the autonomous vehicle must also possess the capability to effectively communicate and coordinate its actions with the manned vehicle.

While many researchers have been discussing autonomous multi-agent operations,^{1,2} more work is needed on how to perform multi-agent health management for autonomous task groups. In the past, the term “health management” was used to define systems which actively monitored and managed vehicle sub-systems (e.g., flight controls, fuel management, avionics) in the event of component failures.³ Prognostic and health management techniques are being developed for new military aircraft systems to reduce future operating, maintenance, and repair costs.⁴ In the context of multiple vehicle operations, we can extend this definition to autonomous multi-agent teams. In this case, teams involved in a mission serve as a “vehicle system.” Then, each multi-agent team involved in the mission is a sub-system to the larger mission team. In addition, each vehicle is a sub-system of each multi-agent team, and so on.

As with mission-critical systems for a single agent, multi-agent task allocation and mission managements systems must account for vehicle- and system-level health-related issues to ensure that these systems are cost effective to operate. For example, despite on-going development of health management techniques for flight-critical systems, most UAVs are controlled by a team of operators from a remote location.⁵ Here, we define an operator as a human being that monitors or issues commands to a UAV during a mission. Although recent advances in mission systems have reduced the number of operators per vehicle, the vehicle to operator ratio for most UAV mission platforms remain less than or equal to one. This ratio is small for many reasons. First, most UAVs are piloted by humans during take-off, landing and other complex flight tasks.^{6–8} This means that for every UAV, there is at least one UAV pilot–operator. In addition, most UAV have remote ground stations to monitor its flight critical systems (e.g., communications link, flight control, guidance/navigation systems) and mission data. Since it is difficult for a pilot–operator to monitor all mission critical information, most UAVs have more than one operator during a mission.⁹ Therefore, in a rapidly changing environment an operator can easily find themselves overloaded with information from multiple UAV teams.

To investigate this problem, we have constructed a unique indoor multi-vehicle testbed to study long duration missions in a controlled environment. This testbed is being used to implement and analyze the performance of techniques for embedding the fleet and vehicle health state into the mission and UAV planning. In particular, we are examining key research questions related to vehicle and multi-agent health management issues, such as vehicle failures, refueling and maintenance using real hardware. The testbed is comprised of aerial and ground vehicle components, allowing researchers to conduct tests for a wide variety of mission scenarios. This paper describes the components/architecture of the testbed and presents recent flight test results.

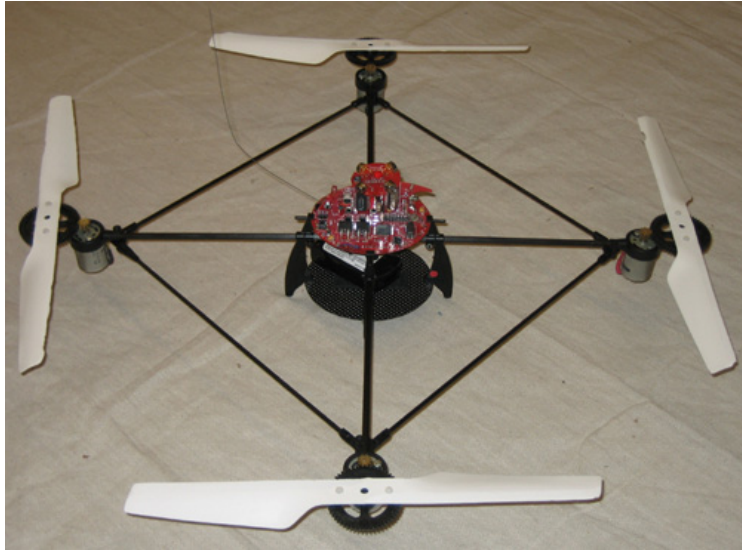


Figure 1: Draganflyer V Ti Pro¹⁵ model for flight testing

II. Background

Research groups have used a variety of platforms to verify advanced theories and approaches in the development of innovative UAV concepts. The BERkeley AeRobot (BEAR) project features a fleet of commercially available rotary-wing and fixed-wing UAVs that have been retrofitted with special electronics. These vehicles have been used in applications such as autonomous exploration in unknown urban environments or probabilistic pursuit-evasion games.^{10,11} At MIT's Aerospace Controls Laboratory (ACL), a testbed consisting of a fleet of eight autonomous unmanned UAVs was designed as a platform for evaluating autonomous coordination and control algorithms.^{12,13} This testbed has been used to compare various control approaches to accomplish autonomous and coordinated missions. Similarly, researchers in the Multiple Agent Intelligent Coordination and Control (MAGICC) Lab at BYU have built and flown a group of small fixed-wing UAVs to perform multi-vehicle experiments outdoors.¹⁴ These planes are launched by hand and track waypoints autonomously.

Likewise, the DragonFly project at Stanford University's Hybrid Systems Laboratory heavily modified a fixed-wing, 10-foot wingspan model aircraft, and a second and third similar aircrafts are under development.¹⁶ The objective of this platform is to provide an inexpensive capability of conducting UAV experimental research, ranging from low-level flight control algorithm design to high-level multiple aircraft coordination. Similarly, in order to demonstrate new concepts in multi-agent control on a real-world platform, the Hybrid Systems Lab developed the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC). STARMAC is a multi-vehicle testbed consisting of two Quad Rotor UAVs that autonomously track a given waypoint trajectory.¹⁷ This platform was selected based on the convenient handling characteristics of the aircraft, their low cost, and their easiness to be modified.

In addition, there are a number of indoor multi-vehicle testbeds that have been built to study

multi-agent activities. For example, the HOVercraft Testbed for DEcentralized Control (HOTDEC) Platform at UIUC is a multi-vehicle ground testbed used for multi-vehicle control and networking research.¹⁸ Researchers at Vanderbilt University have built the Vanderbilt Embedded Computing Platform for Autonomous Vehicles (VECPAV) which has been used to fly two vehicles in an indoor environment.¹⁹ Also, the UltraSwarm Project at the University of Essex is designed to use indoor aerial vehicles to examine questions related to flocking and wireless cluster computing.²⁰

The testbeds listed above have several limitations that inhibit their utility for investigating the health management of UAV teams performing large-scale missions over extended periods of time. For example, outdoor platforms can only be tested during the proper weather and environmental conditions. Since almost all of these UAVs can only be flown safely during daylight operations, these systems cannot be used to examine research questions related to long duration missions at night. In addition, many of these vehicles are modified to carry additional vehicle hardware for flight operations. As a result, these vehicles have to be redesigned to meet payload, on-board sensing, power plant and other requirements. Thus, during flight operations, many of these vehicles must be flown in specific environmental conditions to avoid damage to the vehicle hardware – and this may be unrelated to flight hour constraints. These external UAVs also typically require a large support team, which makes long-term testing logistically difficult and extremely expensive. Furthermore, many of the indoor testbeds are either 2D or operate in 3D in a very limited flight volume.

In contrast, this testbed is uniquely designed to test and examine a wide variety of multi-vehicle missions. Currently, we have demonstrated many types of multi-vehicle coordinated test flights (using both autonomous ground and air vehicles). Since this is an indoor testbed that uses small, unmodified electric helicopters (Draganfly Quadrotors¹⁵ in Figure 1), we have been able to fly more than three air vehicles in a typical-sized room, and it takes no more than one operator to set up the platform for flight testing at any time of day for at period of time. At the heart of the testbed is a global metrology system that yields very accurate, high bandwidth position and attitude data for all vehicles in the entire room. Since the position markers are lightweight, the position system is able to sense vehicle position and attitude without adding any weight to the vehicles. As a result, our testbed configuration does not require modifications to off-the-shelf radio-controlled vehicle hardware, thus the platform’s air vehicles are not over-stressed during flight operations and the testbed is extremely robust. This means that we can perform a very large number of test flights in a short period of time with little logistical overhead. We can also perform joint land and air operations to investigate very realistic mission scenarios. Thus, this platform is ideal for the **rapid prototyping** of multi-vehicle mission management algorithms since it can be operated over long periods of time using one person at a fraction of the cost of what would be needed to support an external flight demonstration.

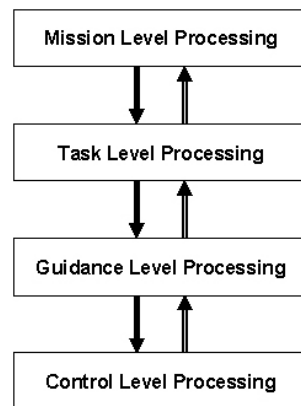


Figure 2: Global architecture model for the indoor testbed

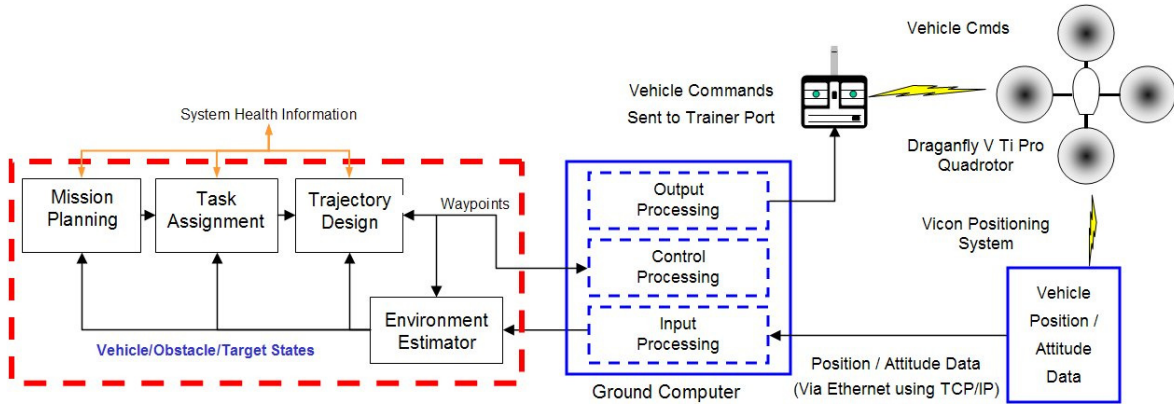


Figure 3: Multi-vehicle command and control architecture block diagram

III. System Architecture and Components

Since the focus of this testbed is to enable researchers to test a variety of multi-vehicle related algorithms in a real-time environment, our goal was to develop a system architecture that allows users to easily add/remove components. Since an additional goal is to develop a system that allows one operator to command a number of vehicles, it is also important that any architecture that is used includes components that will enable an operator to understand what the system is doing at a high-level (i.e., where the vehicles are going, what mission tasks they are pursuing) and at times, issue changes or verifications to the mission plan in accordance with their goals.

One system architecture which was designed to meet these requirements was debuted in the DARPA SEC Capstone flight demonstration by the MIT demonstration team.^{21,22} The goal of this system architecture was to provide a pilot and/or weapons systems officer on board an aircraft with the capability to issue commands to and receive feedback from a UAV performing a mission in a real-time environment. In this system, the pilot and/or weapons systems officer would issue commands to the UAV in English, which would be translated by a Natural Language Interface into commands the vehicle could understand in order to carry out its mission. As discussed in [22], this was the first system to command a UAV using mission level commands through a natural language interface in a real-time, evolving environment, and the basic principles behind the architecture's structure were critical in allowing each system component to be developed independently and assembled efficiently for the June 2004 test flight in less than eight months.

The testbed architecture follows a simple hierarchical design (as shown in Figure 2). The basic premise behind this architecture is as follows: in order to divide up the workload on any individual system, each component in the architecture must be able to communicate effectively with the other components. Therefore, as long as the interface requirements between each system component are well-defined and each component meets its system requirements, the component can be substituted easily by another component which meets the architecture's interface and system requirements (although its operation details may be quite different than the component it replaced) – thus,

allowing users to add and remove components as needed. This design approach has been used by many people for designing complex systems architectures.

As shown in Figure 3, this system architecture has four major components: a mission planning level designed to set the system goals and monitor system progress, a task assignment level which (in general) is designed to issue and assign specific tasks to a vehicle or vehicle group in order to support the overall mission goals, a trajectory design level which directs each vehicle and its subsystems on how to best perform the actual tasks provided by the task processing level, and a control processing level designed to carry out the activities set by upper-levels in the system. Note that health information about each component in the system is provided to and used by each component in the architecture in the decision making process. Similar architectures have been used in other multi-vehicle testbeds. For example, Ref. [13] uses a similar architecture to coordinate multiple vehicles to carry-out a high-level mission plan provided by an external source. Each vehicle uses a receding horizon task assignment (RHTA) algorithm (at the task processing level) which allows the vehicle to coordinate and carry out its task goal with other vehicles in the multi-vehicle team, a receding horizon trajectory planner (at the guidance level) which formulates the guidance problem as a mixed integer linear program that can be solved in real-time, and a control architecture (at the control level) to implement the vehicle’s task and guidance plans. Each component is designed to support the decisions made in each level to ensure that the vehicle makes an informed decision which is in the team’s best interest for any given task.

As part of this testbed, our focus is to develop health monitoring systems for each of these components. The health monitoring components are designed to evaluate the performance of each component and provide feedback to the rest of the system (and operator) regarding its progress and mission effectiveness. This information will be used by the rest of the system to potentially adjust or redirect its mission and/or task goals as the mission is taking place.

IV. Testbed Hardware

In order to test and demonstrate the real-time capabilities of these health management algorithms in a realistic real-time environment, we sought to develop a low-cost, indoor testing environment which could be used over extended periods of time in a controlled environment. First, the Draganflyer V Ti Pro¹⁵ (as shown in Figure 1) was selected for this testbed for a number of reasons. The Draganflyer is a small (approx. 0.7 m from blade tip to blade tip), lightweight (under 500 g) air vehicle with a payload capacity of about 100 g that can fly between 13–17 mins on one battery charge (using a 2000 mAh battery) while carrying a small camera. The four-bladed design helps to simplify the vehicle dynamics during flight operations, while the vehicle’s airframe is very robust and extremely easy to fix in the event of a crash. In addition, the rotor blades are designed to fracture when they hit a solid object (thus preserving an indoor environment). Thus, the vehicle is designed to be durable and safe – making it suitable for an indoor flight testbed.

Figure 4 shows a diagram of the components and setup of the integrated system. Note that all computing for this system is done on ground-based computers, which have two AMD 64-bit Opteron processors, 2 Gb of memory and run Gentoo Linux. The control processing and command

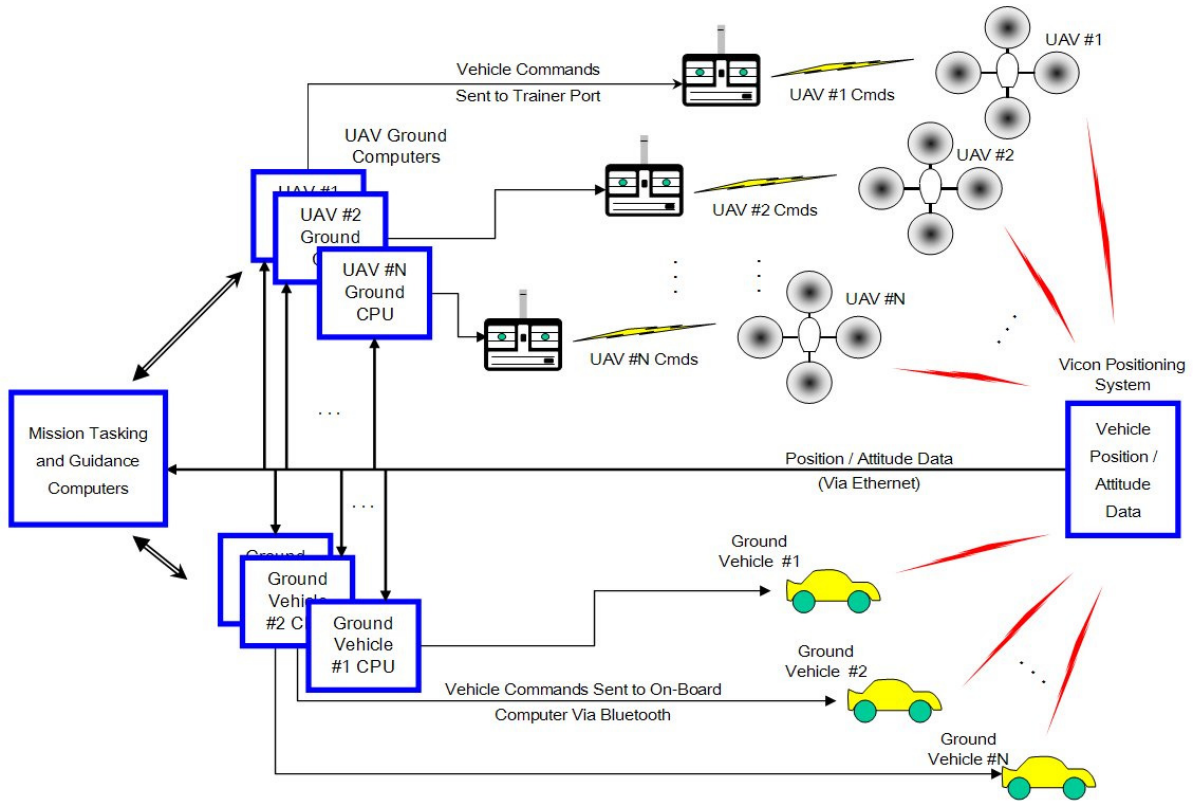


Figure 4: Integrated vehicle system block diagram

data is processed by this computer and sent over an RS-232 connection from the vehicle's control computer to the vehicle's R/C Transmitter over the transmitter's trainer port interface.

A Vicon MX camera system²³ is used to detect the vehicle's position and orientation in real-time. By attaching lightweight reflective balls to the vehicle's structure, the Vicon MX Camera system and Tarsus software can track and compute the vehicle's position and attitude information up to 120 Hz with a 10 msec delay. This data is then transmitted via ethernet using TCP/IP to the vehicle's ground based control computer.

As described above, no structural or electronics modifications were made to the Draganflyer V Ti Pros used for this testing. This allows us to autonomously operate unmodified R/C vehicles without requiring any additional battery or computer payload be carried. This helps maximize the vehicle flight time and reduces stress on the motor/blade components.

It is very difficult to confirm the sensor accuracy in flight, but Figure 5 shows a scatter plot of the measured (x,y) position (in m) of a quadrotor sitting on the floor at position $(0,0)$. Note the scale on the plot – with the rotors not turning, the maximum x -position error measured by the system is 0.000325 m and the maximum y -position error measured by the system is 0.000199 m. Tracking multiple balls in a unique orientation on each vehicle enables the Vicon system to determine the position of the center of mass and the attitude of each air/ground vehicle that is within range, which is on the order of tens of meters. Our current six camera configuration can easily track at

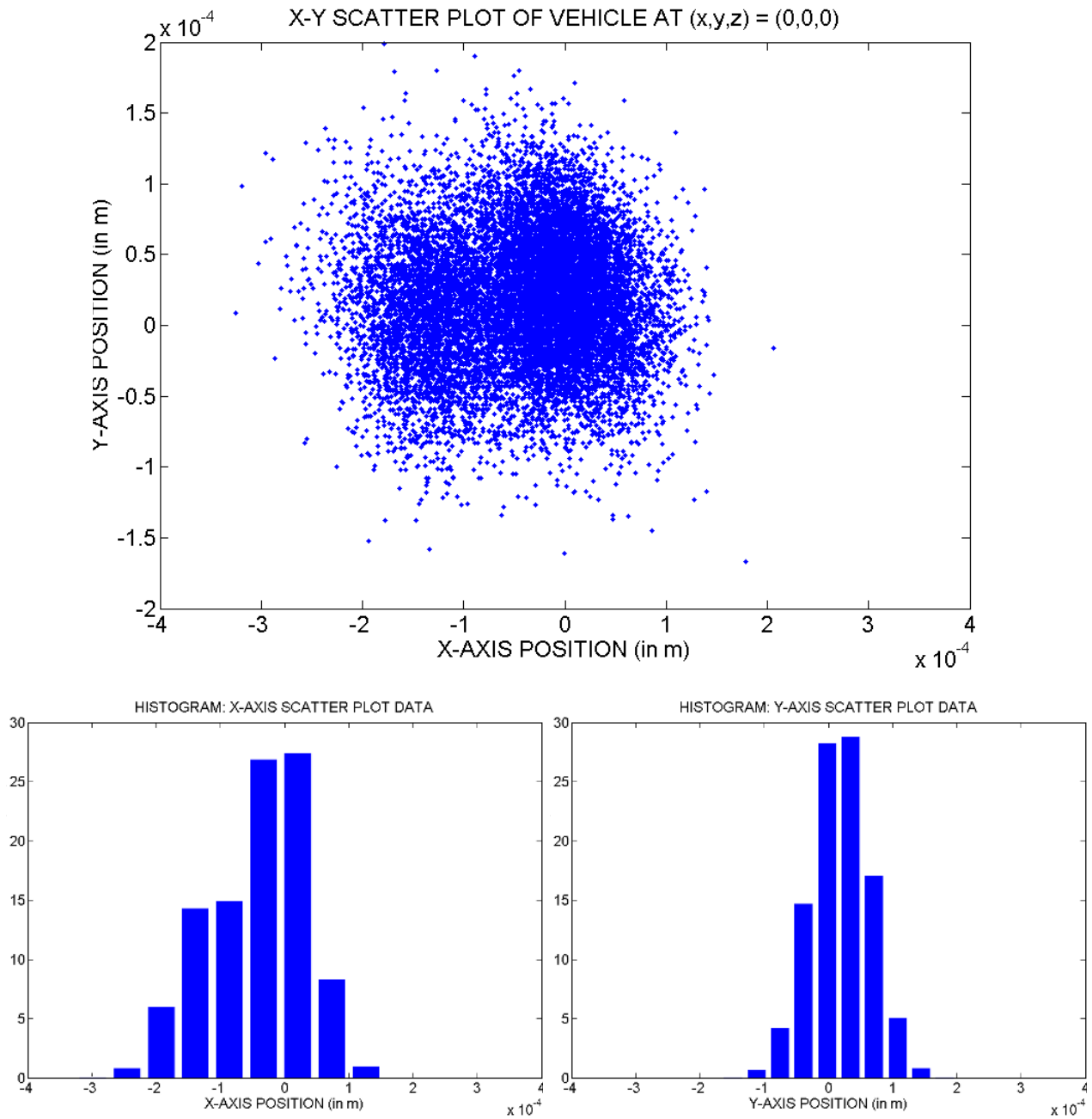


Figure 5: Scatter plot of (x, y) vehicle position - rotors not turning (top), histograms with percentage of time at location for x-, and y-positions (below). Note that the scale in these plots is meters $\times 10^{-4}$

least four air vehicles and multiple ground vehicles in a 5 m by 5 m by 2 m flight volume.

A vehicle model was needed in order to control the vehicle. Starting with the generic quadrotor vehicle model developed in [24]

$$\begin{aligned}
\ddot{x} &= (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{1}{m} U_1 \\
\ddot{y} &= (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{1}{m} U_1 \\
\ddot{z} &= -g + (\cos \phi \cos \theta) \frac{1}{m} U_1 \\
\ddot{\phi} &= \dot{\theta} \dot{\psi} \left(\frac{I_y - I_z}{I_x} \right) - \frac{J_R}{I_x} \dot{\theta} d + \frac{L}{I_x} U_2 \\
\ddot{\theta} &= \dot{\phi} \dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) + \frac{J_R}{I_y} \dot{\phi} d + \frac{L}{I_y} U_3 \\
\ddot{\psi} &= \dot{\phi} \dot{\theta} \left(\frac{I_x - I_y}{I_z} \right) + \frac{1}{I_z} U_4
\end{aligned} \tag{1}$$

where x, y, z are the vehicle's position variables, $\phi, \theta,$ and ψ are the roll, pitch and yaw orientations of the vehicle (respectively), $I_x, I_y,$ and I_z are the body moment of inertias, L is the length between the vehicle CG and a motor, m is the mass of the vehicle, g is the gravitational constant, J_R is the moment of inertia of a rotor blade, Ω represents a disturbance generated by differences in rotor speed, and $U_1, U_2, U_3,$ and U_4 are the collective, roll, pitch and yaw input commands (respectively) for the vehicle. After linearizing this model, setting $U_1 = m g + \delta u_1$ and dropping small terms in the x and y dynamics, yields

$$\begin{aligned}
\ddot{x} &= g\theta, & \ddot{\phi} &= \frac{L}{I_x} U_2 \\
\ddot{y} &= -g\phi, & \ddot{\theta} &= \frac{L}{I_y} U_3 \\
\ddot{z} &= \frac{1}{m} \delta u_1, & \ddot{\psi} &= \frac{1}{I_z} U_4
\end{aligned} \tag{2}$$

which is the simplified linearized vehicle model around the hover flight condition. The controller architecture is broken into four groups (i) $\theta \rightarrow x$, (ii) $\phi \rightarrow y$, (iii) z and (iv) ψ .

An integrator of the position / heading error was added to the model of each loop so that the controller would remove steady state position /heading errors in hover. A simple LQR controller was developed and used to control each loop. These controllers were designed to ensure that the vehicle could quickly respond to position errors, while minimizing overshoot. This design was used mainly because the platform will be used in the future for surveillance experiments. In these experiments, a camera will be mounted on the vehicle facing toward the ground – thus, large changes in pitch and roll will affect the vehicle's ability to focus on an item on the ground during surveillance activities. Note that the $\phi \rightarrow y$ loop is nearly identical to the $\theta \rightarrow x$ loop because of vehicle's symmetry.

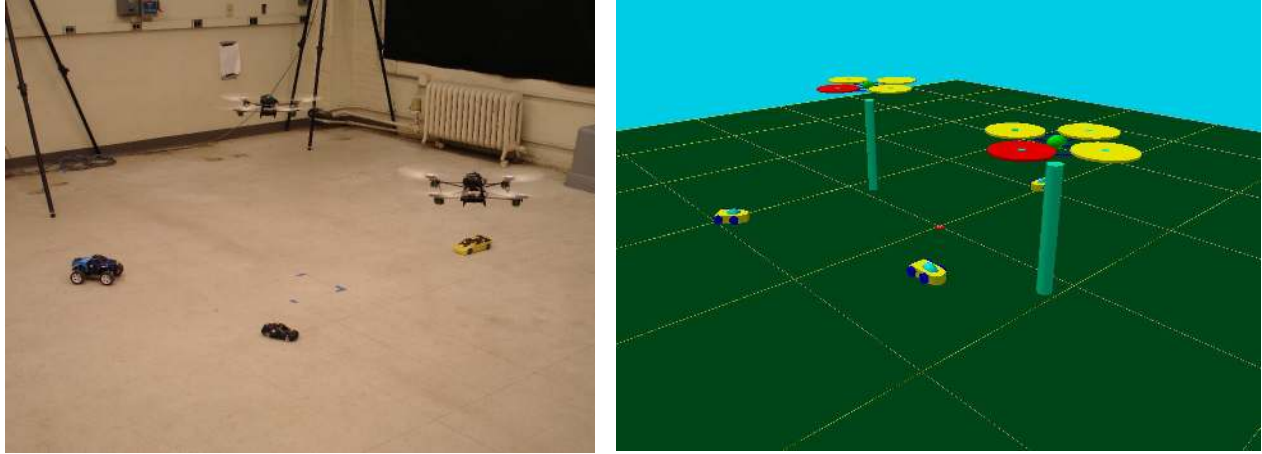


Figure 6: Multi-vehicle search and track experiment (left) and operator interface visualization (right). The sensing system records the ground vehicle locations in real-time, and if the vehicles are being tracked by the UAVs, then the location is displayed to the operator

V. Task Processing and Operator Interface Components

As discussed in the previous section, each vehicle is capable of processing and implementing tasks defined by a system component or user. For example, each vehicle has a vehicle manager module designed to handle task processing, trajectory generation and control processing for the vehicle. This module is designed to allow an external system or user to communicate with the vehicle using task-level commands, such as (but not limited to):

- Fly to Waypoint A
- Hover / Loiter over Area B
- Search Region C
- Classify / Assess Object D
- Track / Follow Object E
- Take-off / Land at Location F

These commands can be sent to the vehicle at any time during vehicle operations. Each agent's vehicle manager will process these tasks as they arrive and respond to the sender acknowledging the task request.

In addition, the testbed is designed with an automated system task manager. Since each air vehicle in the system can take-off and land autonomously, this task manager can autonomously manage any air and ground vehicle controlled by the system using these task level commands. As a result, multi-vehicle mission scenarios (e.g., search, persistent surveillance, and others) can be organized and implemented by the task manager autonomously. Likewise, coordinated multi-vehicle flight tasks can also be managed by the task advisor with very little operator interaction with the system, thus allowing one operator to command and control many vehicles in the testbed

during a mission. In fact, the task manager has been used to successfully conduct multi-vehicle persistent surveillance mission tests using two vehicles and lasting over 20 mins. In these tests, the task manager autonomously commanded one vehicle to hover over a specified location in the flight area, and after five minutes commanded the other vehicle (which was waiting on the ground) to cycle in and replace the flying vehicle. Then, after each vehicle landed, it received a new battery from the operator. Therefore, after the test was started, the operator's only role was to change out the battery of the vehicle which just landed since the tasking system managed the entire test autonomously. This test was successfully repeated multiple times.

Although this system reduces the operator load by handling many tasks autonomously, the system is designed with an operator interface with vehicle tasking capability. The task manager system is designed to allow an operator to issue a command to any vehicle (at any time). Currently, the operator interface includes 3D display of the objects in the testing area (as shown in Figure 6) and a command and control GUI, which displays vehicle health and state data, task information, and other mission-relevant data.

VI. Results

Various components of this indoor multi-vehicle testbed have been under development since May 2005. The goal is to study long duration missions in a controlled environment, so the recent focus has been to ensure that the testbed can *reliably* fly multiple mission sorties. Therefore, while each vehicle must be able to reliably perform the tasks it has been issued, it must also be able to move throughout the indoor environment in a predictable way. As shown in Figure 8, we have been flying a variety of multi-vehicle tests and mission scenarios with the testbed. Since January 2006, we have flown over 500 vehicle experiments, including over 60 flight demonstrations (more than 30 per day) over a 16 hour period at the Boeing Technology Exposition at Hanscom AFB on May 3rd and 4th, 2006. Each of the tests performed at the event involved two vehicles – one test involved two air vehicles flying a 3D coordinated pattern (as shown in Figure 11), the other involved an air vehicle following a ground vehicle. These demonstrations showed that the platform is capable of flying multiple UAV missions repeatedly, on-demand, and with minimal setup.

Typical results from a 10-minute hover test are shown in Figure 9. In this test the vehicle was commanded to hold its position at $(x, y, z) = (0, 0, 0.75)$ m for a 10-minute period of time. Note that there were no other vehicles flying in the room during this test flight. Figure 9 shows four plots, including a plot of the vehicle x - y location while it maintained its position. The dashed red box in the picture is at ± 10 cm from the center point, and it is clear that for nearly the entire flight the vehicle maintained its position inside this 20 centimeter box. The remaining plots in the figure are the histograms of the vehicle's X-, Y-, and Z-Position during these tests. Note that the vehicle maintained its altitude (staying between 0.7 and 0.8 meters) during the entire hover test.

The results of a single-vehicle waypoint tracking experiment are shown in Figure 10. In this test, the vehicle was commanded to take-off at $(x, y, z) = (0, 0, 0)$ m and fly to $(0, 0, 1)$ m. Then, the vehicle was commanded to fly to and hover (for 10 seconds) at each of the following waypoints: $(-1, 0, 1)$, $(-1, -1, 1)$, $(1, -1, 1)$, $(1, 1, 1)$, $(-1, 1, 1)$ at a commanded velocity of 0.05 m per second. After

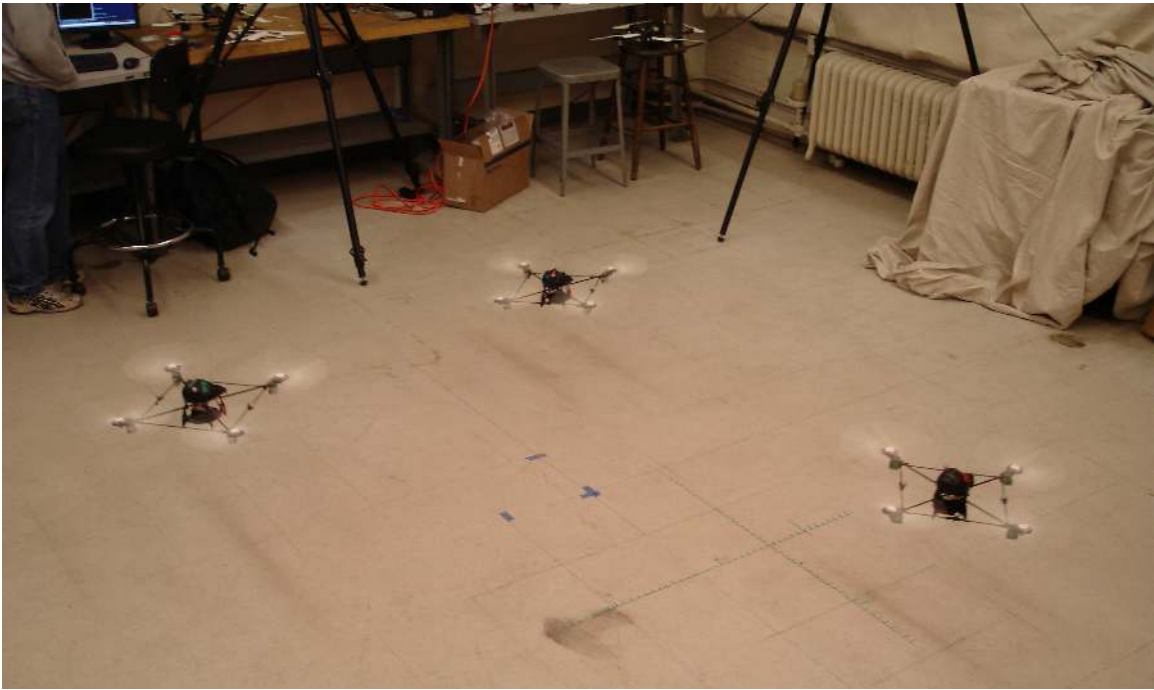


Figure 7: Fully autonomous flight test with three UAVs



Figure 8: Close-up on the UAVs in flight

reaching the final waypoint, the vehicle was commanded to land at $(x, y, z) = (-1, 1, 0)$ m. The purpose of this test was to observe the vehicle as it tries to follow a set trajectory. The plots show that the vehicle followed the trajectory as specified. The cross-track error observed was less than 15 cm from the specified trajectory at any given time during the flight. Note that position errors near $(x, y, z) = (-1, 1, 1)$ in both plots are a result of the vehicle's automated landing sequence. When an air vehicle in the system goes into an automated landing, it is commanded to hover 0.5 m above the floor (which is just above the vehicle's ground effect) before executing its final descent. This is to minimize the vehicle's position error during landing (resulting from ground effect). Also, note that in both plots the vehicle overshoots each of the specified waypoint position due to the fact that vehicle's current waypoint follower does not cause the vehicle to flair early enough to slow the vehicle down before reaching the waypoint. As a result, the current waypoint follower is being upgraded to reduce this error during flight.

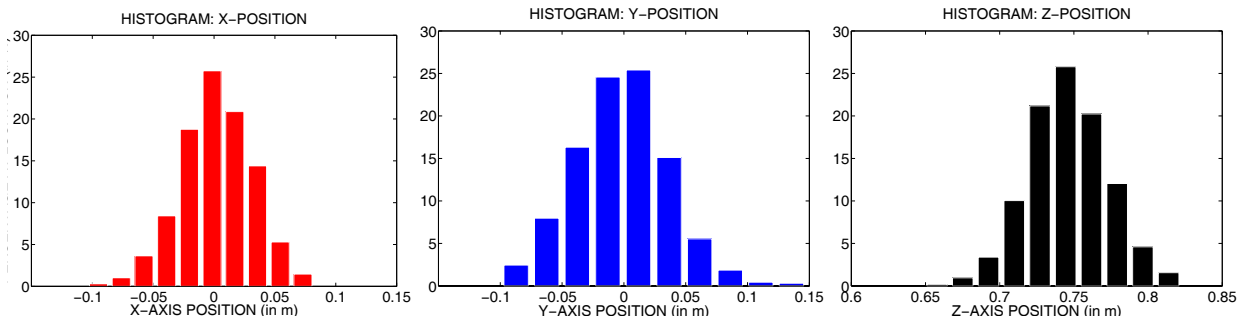
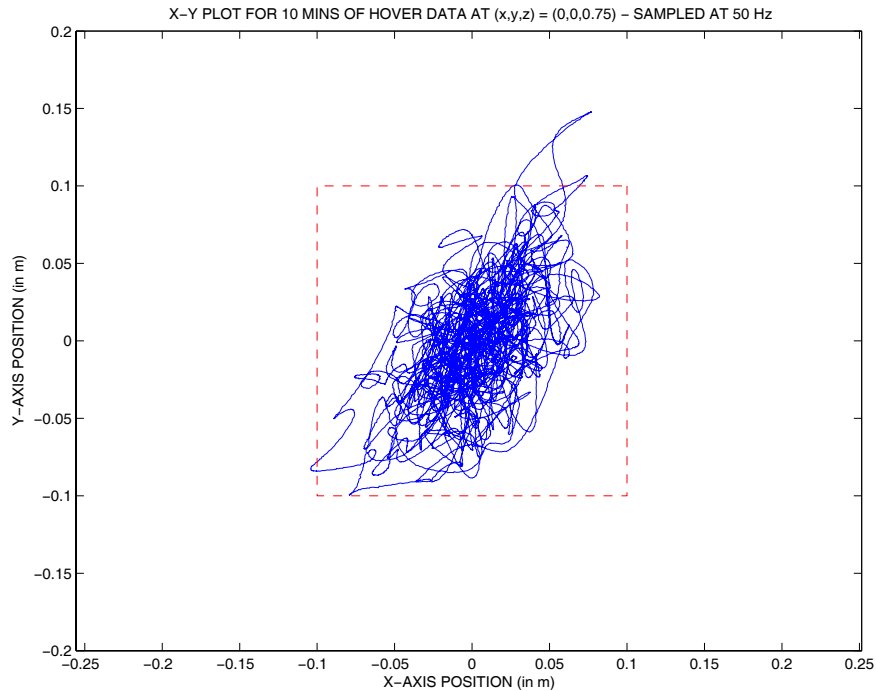
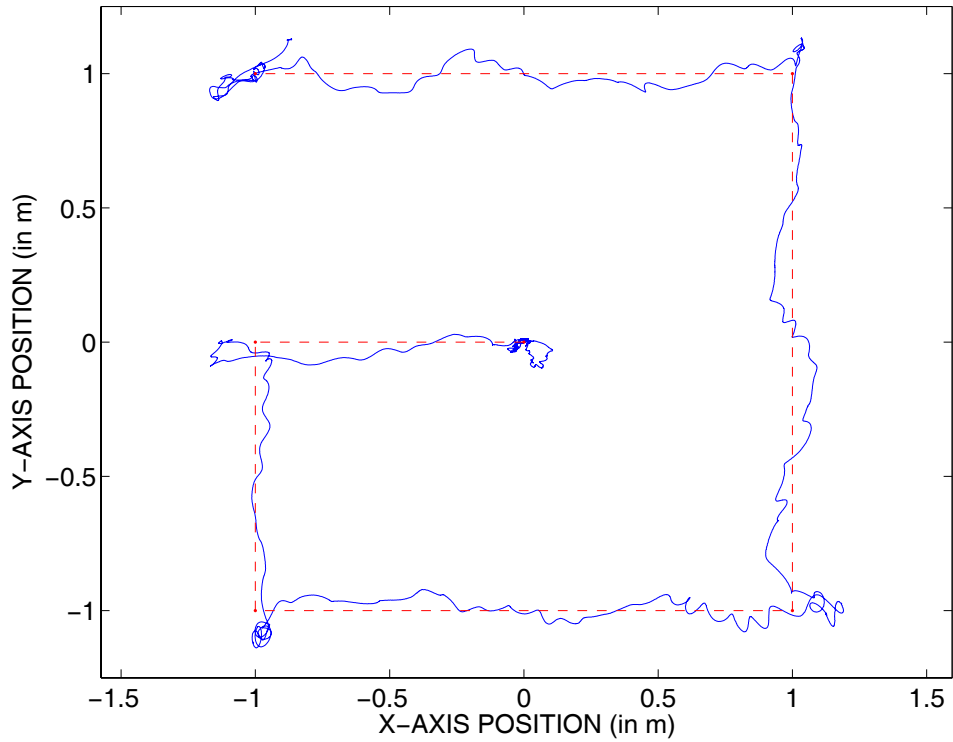


Figure 9: Single vehicle hover experiment – UAV commanded to hover at $(x, y, z) = (0, 0, 0.75)$ m for 10 min: x - y plot of vehicle position (top), histograms with percentage of time at location for x , y , and z positions (below)

In addition to these single-vehicle experiments, several multi-vehicle experiment and test scenarios have been conducted. These tests include, but are not limited to, formation flight tests, coordinated vehicle tests (involving three air vehicles), and multi-vehicle search and track scenarios. Figure 11 shows the results from a two-vehicle coordinated flight experiment. In this experiment, the vehicles were commanded by the system’s task advisor to take-off and fly a circular trajectory maintaining a constant speed of 0.25 m per second and 180 degrees of angular separation. Note that in this test, not only are the vehicles flying in a circle (as projected in the x - y coordinate frame), but they are also changing altitude as they fly around the pattern – going from an altitude of 0.5 m to 1.5 m as the vehicles fly around the pattern.

Figure 11 shows three plots. The upper left plot shows the x - y projection of one of the five circle test flights that was completed as part of this experiment. Notice that the vehicle trajectories

X-Y PLOT OF PATTERN FLIGHT: TEST 1 (INCLUDES AUTO TAKE-OFF / LANDING)



X-Y PLOT OF PATTERN FLIGHT: TEST 2 (INCLUDES AUTO TAKE-OFF / LANDING)

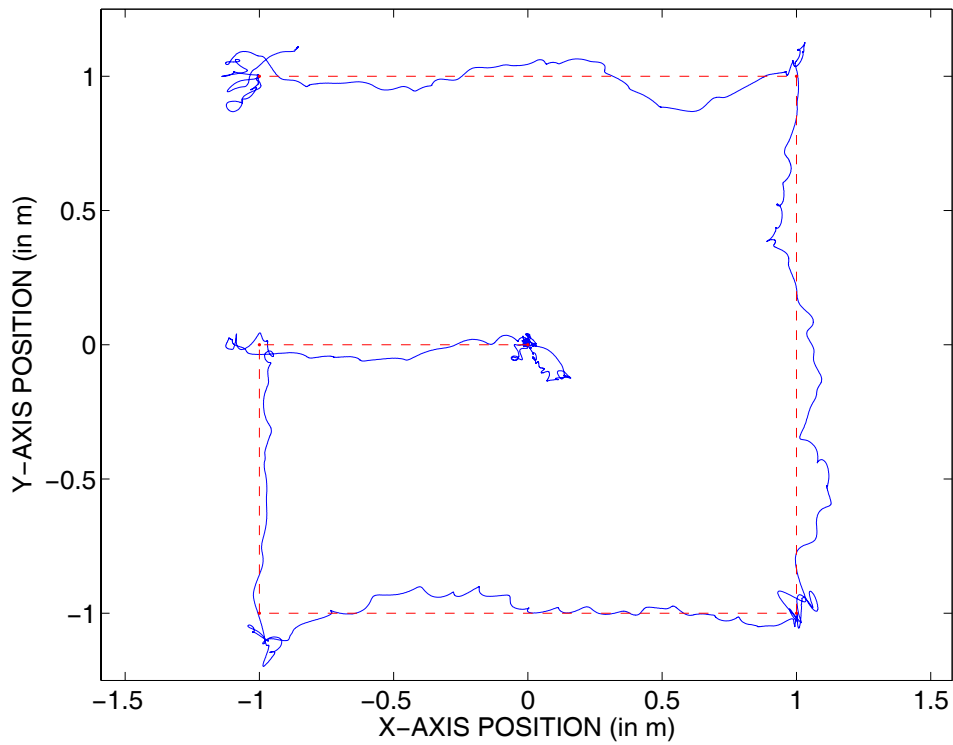


Figure 10: Single vehicle waypoint tracking experiment: Vehicle takes-off at (0,0,1) m and then flies to points (-1,0,1), (-1,-1,1), (1,-1,1), (1,1,1), (-1,1,1) and lands at (-1,1,0) m

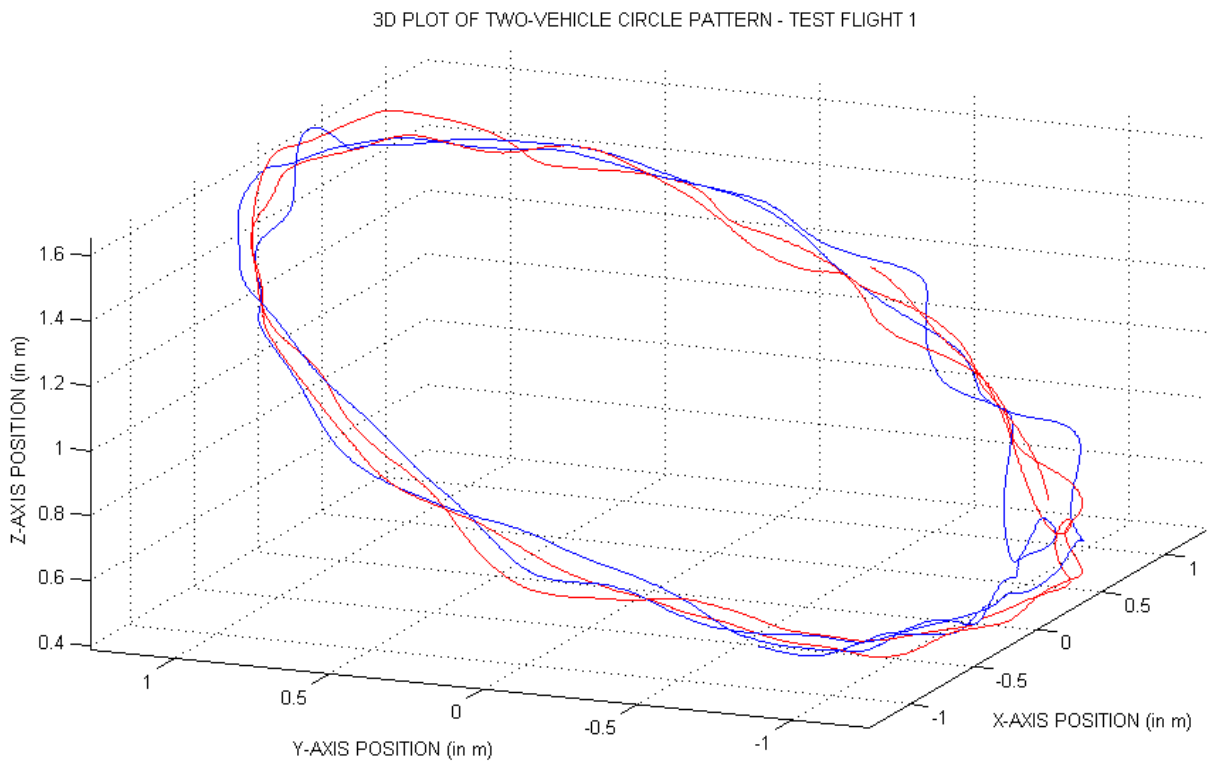
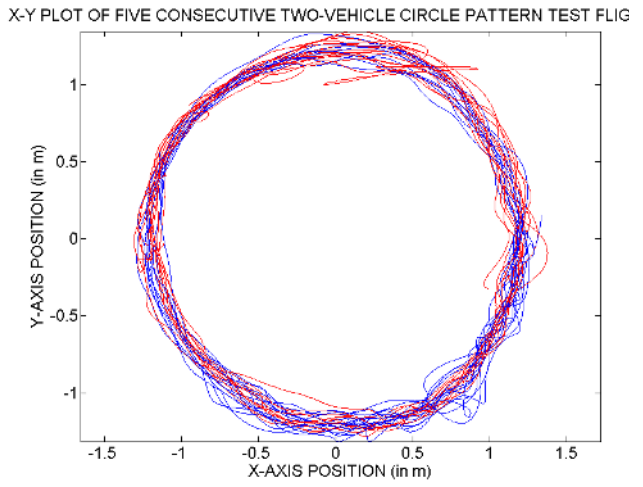
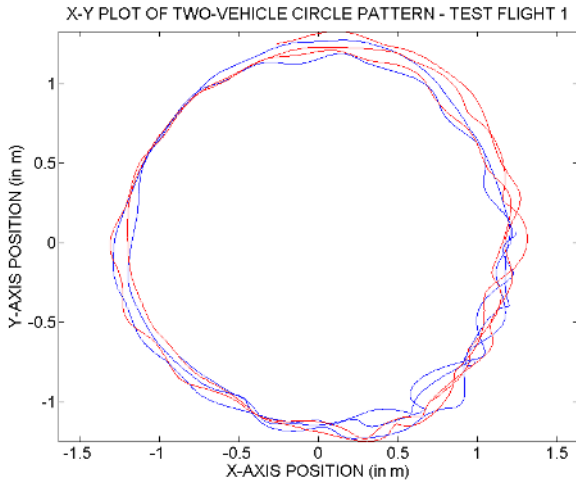


Figure 11: Multi-vehicle coordinated flight experiment: Two-vehicles flying at a constant speed in a circular pattern with changes in altitude

in the lower right corner of the plot appear to be more noisy. This disruption is partially caused by the fact that, when each vehicle is flying through this region, they are trying to increase in altitude through the rotor downwash from the previous vehicle. We have found in our testing that the rotor downwash for these quad rotor vehicles is substantial, thus making it very difficult to fly a vehicle underneath one another without significant altitude separation. The lower plot shows a three-dimensional view of the trajectory, making it clear that the vehicles are also changing altitude during the flight. In addition, the upper right plot shows the results of five consecutive two-vehicle circle test flights. These test flights were performed over a 20 minute time span, demonstrating that these results are repeatable and the platform was capable of performing multiple test flights over a short period of time.

VII. Conclusions and Future Work

This paper described a new indoor multi-vehicle testbed developed at MIT to study long duration missions in a controlled environment. This testbed has enabled an increase to the unmanned vehicle to operator ratio for multi-vehicle missions to 4:1, which is an important achievement in the overall goal of reducing the cost and logistic support needed to operate these systems.

Furthermore, while many testbeds have been developed to investigate the command and control of multiple UAVs, this one is unique in that real aerial vehicles fly autonomously indoors, with virtually no restrictions on their range and orientation. This also represents a significant step forward, and should enable rapid prototyping of coordination algorithms in a controlled environment.

The testbed also has a separate landing and ground maintenance system to support the vehicle hardware in an around-the-clock environment. More specifically, the landing hardware and its associated real-time processing is used to aid the vehicle's local guidance and control modules during the take-off and landing tasks. In addition, a maintenance module will be used to evaluate whether the actual vehicles are due for maintenance and monitor the recharging of the batteries prior to flight. We have successfully demonstrated this concept, and are currently improving our design prior to fully integrating it into the testbed.

Finally, algorithms to determine the health of each mission component in real-time have been successfully implemented and tested. The vehicles have an onboard software monitor that determines when a vehicle must return to its base location for recharging. This, and other health management information about each component, are currently being used in the mission system to enable the strategic and tactical level planners to make informed decisions about the best way to allocate resources given the impact of likely failures.

Acknowledgements

The authors would like to thank Glenn Tournier, Michael Robbins, Daniel Dale, Ray He and Erwan Salaun for their invaluable assistance in the project. Brett Bethke would like to thank the Hertz Foundation and the American Society for Engineering Education for their support of this research. This research has been generously supported by the Boeing Company under the guidance

of Dr. John Vian at the Boeing Phantom Works, Seattle. Research also funded in part by AFOSR grant FA9550-04-1-0458.

References

- ¹Gaudio, P., Shargel, B., Bonabeau, E., and Clough, B., "Control of UAV SWARMS: What the Bugs Can Teach Us," *Proceedings of the 2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations Aerospace Conference*, San Diego, CA, September 2003.
- ²Paruanak, H., Brueckner, S., and Odell, J., "Swarming Coordination of Multiple UAV's for Collaborative Sensing," *Proceedings of the 2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations Aerospace Conference*, San Diego, CA, September 2003.
- ³Fudge, M., Stagliano, T., and Tsiao, S., "Non-Traditional Flight Safety Systems and Integrated Vehicle Health Management Systems," Produced for the Federal Aviation Administration, ITT Industries, Advanced Engineering and Sciences Division, 2560 Alexandria Drive, Alexandria, VA 22303, August 2003.
- ⁴Becker, K. C., Byington, C. S., Forbes, N. A., and Nickerson, G. W., "Predicting and Preventing Machine Failures," *The Industrial Physicist*, Vol. 4, No. 4, December 1998, pp. 20–23.
- ⁵Prieditis, A., Dalal, M., Arcilla, A., Groel, B., Bock, M. V. D., and Kong, R., "SmartSwarms: Distributed UAVs that Think," *2004 Department of Defense Command and Control Research and Technology Symposium*, San Diego, CA, 2004.
- ⁶Frew, E., Spry, S., McGee, T., Xiao, X., Hedrick, J. K., and Sengupta, R., "Flight Demonstrations of Self-Directed Collaborative Navigation of Small Unmanned Aircraft," *Proceedings of the AIAA 3rd Unmanned Unlimited Technical Conference, Workshop, and Exhibit*, Chicago, IL, September 2004.
- ⁷Gavrilets, V., Frazzoli, E., Mettler, B., Piedmonte, M., and Feron, E., "Aggressive Maneuvering of Small Helicopters: A Human Centered Approach," *International Journal of Robotics Research*, Vol. 20, October 2001, pp. 705–807.
- ⁸Gavrilets, V., Martinos, I., Mettler, B., and Feron, E., "Flight Test and Simulation Results for an Autonomous Acrobatic Helicopter," *Proceedings of the AIAA/IEEE Digital Avionics Systems Conference*, Irvine, CA, October 2002.
- ⁹McCarley, J. S. and Wickens, C. D., "Human Factors in UAV Flight," Produced for the Federal Aviation Administration, Human Factors Research and Engineering Division – 2004 Annual Report, Institute of Aviation, Aviation Human Factors Division, University of Illinois at Urbana-Champaign, 2560 Alexandria Drive, Alexandria, VA 22303, 2004.
- ¹⁰Shim, D., Chung, H., Kim, H. J., and Sastry, S., "Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles," *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA, August 2005.
- ¹¹Vidal, R., Shakernia, O., Kim, H. J., Shim, H., and Sastry, S., "Multi-Agent Probabilistic Pursuit-Evasion Games with Unmanned Ground and Aerial Vehicles," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, 2002, pp. 662–669.
- ¹²How, J. P., King, E., and Kuwata, Y., "Flight Demonstrations of Cooperative Control for UAV Teams," *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, Chicago, IL, September 2004.
- ¹³King, E., Alighanbari, M., Kuwata, Y., and How, J. P., "Coordination and Control Experiments on a Multi-Vehicle Testbed," *Proceedings of the IEEE American Control Conference*, 2004.
- ¹⁴Nelson, D. R., Barber, D. B., McLain, T. W., and Beard, R. W., "Vector Field Path Following for Small Unmanned Air Vehicles," *Proceedings of the 2006 American Control Conference*, Minneapolis, MN, June 2006.
- ¹⁵Draganfly Innovations Inc., "Draganfly V Ti Pro Website," <http://www.rctoys.com/draganflyer5tipro.php>, January 2006.
- ¹⁶Evans, J., Inalhan, G., Jang, J. S., Teo, R., and Tomlin, C., "DragonFly: A Versatile UAV Platform for the

Advancement of Aircraft Navigation and Control,” *In the Proceedings of the 20th IEEE Digital Avionics Systems Conference*, October 2001.

¹⁷Hoffmann, G., Rajnarayan, D. G., Waslander, S. L., Dostal, D., Jang, J. S., and Tomlin, C., “The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (STARMAC),” *In the Proceedings of the 23rd Digital Avionics Systems Conference*, Salt Lake City, UT, November 2004.

¹⁸Vladimerouy, V., Stubbs, A., Rubel, J., Fulford, A., Strick, J., and Dullerud, G., “A Hovercraft Testbed for Decentralized and Cooperative Control,” *Proceedings of the 2004 American Control Conference*, Boston, MA, July 2004.

¹⁹Koo, T. J., “Vanderbilt Embedded Computing Platform for Autonomous Vehicles (VECPAV),” <http://www.vuse.vanderbilt.edu/kootj/Projects/VECPAV/>, July 2006.

²⁰Holland, O., Woods, J., Nardi, R. D., and Clark, A., “Beyond Swarm Intelligence: The UltraSwarm,” *Proceedings of the 2005 IEEE Swarm Intelligence Symposium*, Pasadena, CA, June 2005.

²¹Valenti, M., Schouwenaars, T., Kuwata, Y., Feron, E., How, J., and Paunicka, J., “Implementation of a Manned Vehicle-UAV Mission System,” *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, RI, August 2004.

²²Schouwenaars, T., Valenti, M., Kuwata, Y., Feron, E., How, J., Roche, E., and Paunicka, J., “Linear Programming and Language Processing Human-UAV Mission Team Mission Specification and Planning,” *Submitted to the AIAA Journal of Guidance, Control and Dynamics*, 2005.

²³Vicon, “Vicon MX Systems,” <http://www.vicon.com/products/viconmx.html>, July 2006.

²⁴S. Bouabdallah, A. N. and Siegwart, R., “Modeling of the “OS” Quadrotor - Version 1.0,” Tech. rep., Ecole Polytechnique Federale de Lausanne, 2004.