

# The Georgia Tech Unmanned Aerial Research Vehicle: GTMax

Eric N. Johnson\* and Daniel P. Schrage†

*School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150*

## ABSTRACT

This paper describes the design, development, and operation of a research Unmanned Aerial Vehicle (UAV) system that has been developed at the Georgia Institute of Technology, called the GTMax. This description will include the processes put in place to enable the system to be used for UAV-technology research, including effective flight testing. Research UAVs are characterized by the need for continual checkout of experimental software and hardware. Also, flight-testing can be further leveraged by complementing research results with flight-test validated simulation results for the same experimental UAV platform. The chosen helicopter-based UAV platform (a Yamaha R-Max) is well instrumented, including: differential GPS, inertial measurement unit, sonar altimeter, radar altimeter, and a 3-axis magnetometer. One or two flight processors can be utilized.

## INTRODUCTION

This paper presents the development of an open system Unmanned Aerial Vehicle (UAV) testbed at the Georgia Institute of Technology and its use in the DARPA Software Enabled Control (SEC) Program. The School of Aerospace Engineering at the Georgia Institute of Technology has been involved in Vertical Takeoff and Landing (VTOL) Unmanned Aerial Vehicle (UAV) technology development for more than ten years. This experience has included initiation of the Association for Unmanned Vehicle Systems, International (AUVSI) International Aerial Robotics Competition and winning it in 1993 with the first demonstration of autonomous flight of an unmanned helicopter, including takeoff and landing, at the contest<sup>1</sup>. This was followed by the U.S. Army Autonomous Scout Rotorcraft Testbed (ASRT) project from 1994 to 1997.

In 1997 Georgia Tech purchased two Yamaha R-50 remotely piloted helicopters (RPHs) for use in flight controls research under the Army/NASA sponsored Georgia Tech Center of Excellence in Rotorcraft Technology (CERT) program. Flight control technologies, such as neural network adaptive flight control, tested on these RPHs have been transferred to Boeing for flight tests on the X-36 and JDAM programs as well as to NASA Ames and MSFC for use in simulation studies<sup>2-5</sup>. Other researchers have also had success utilizing Yamaha helicopters<sup>6-8</sup>.

In 1998 Georgia Tech was awarded a seedling grant under the DARPA SEC Program for research and development of an Open Control Platform (OCP) and for on-line control algorithms<sup>9</sup>. Georgia Tech then acquired a Yamaha R-Max RPH, with twice the payload of the R-50s, for use in the SEC and other research and development programs. Georgia Tech has developed an open system UAV testbed based on this platform. This open system UAV testbed is referred to as the GTMax, Figure 1, and includes four major elements. These are the basic Yamaha R-Max RPH, a modular avionics system, the SEC OCP (being developed jointly with Boeing Phantom Works) with baseline guidance/navigation/control components (software), and a set of simulation tools.



*Figure 1 – GTMax UAV*

In January 2002 Georgia Tech was chosen to be the SEC rotary wing experiments lead. This position includes working with the other SEC technology developers in integrating and demonstrating their SEC technologies on the GTMax. A unique integrated simulation and flight testing approach has been

\* Lockheed Martin Assistant Professor of Avionics Integration, Member AIAA.

E-mail: Eric.Johnson@aerospace.gatech.edu

† Professor

Fellow AIAA.

E-mail: Daniel.Schrage@aerospace.gatech.edu

developed to support these activities. It includes a smooth transition from Software-In-The-Loop (SITL) to Hardware-In-The-Loop (HITL) simulation, followed by flight testing. An SEC benchmark demonstration was completed in May 2002 and a series of planned mid-term and final experiments will be conducted over the next 18 months.

This paper describes the development of the GTMax, and related simulation tools. The GTMax has already been used to accomplish important research objectives and to compete in the international aerial robotics competition<sup>1</sup>. The GTMax system will first be described in some detail. Then, the simulation tools and processes developed to support its development and use will be described. Following this, flight test results and conclusions are presented.

#### GTMax UAV

The GTMax utilizes the Yamaha R-Max industrial helicopter airframe, which has the following characteristics:

- Rotor Diameter: 10.2 ft, Length: 11.9 ft (including rotor)
- Gross Weight: 205 lb, Payload: >66 lb
- Gasoline, 2 Cylinder, Water Cooled, 246cc displacement, 21 hp
- Endurance of approximately 60 min (hover)
- Generator and Battery, 12V
- Electric Starter

The hardware components that make up the basic flight avionics include general purpose processing capabilities and sensing, and add approximately 35 lbs to the basic airframe, depending on configuration. The interface to the helicopter is via a modified Yamaha Attitude Control System (YACS) interface that allows raw servo commands to be given without modification by the stability augmentation. The current configuration includes:

- 266MHz Embedded PC, 500 Mb Flash Drive
- 850 MHz Embedded PC, 500 Mb Flash Drive
- ISIS Inertial Measurement Unit (IMU) (3 accelerometers, 3 rate gyros)
- NovAtel RT-2 Differential GPS (DGPS)
- 3-Axis magnetometer
- Sonar altimeter
- Vehicle telemetry (RPM, Voltage, Remote Pilot Inputs) from YACS

- Actuator control interface to YACS
- 11 Mbps Ethernet data link and an Ethernet switch
- RS-232 serial data link
- Axis video server and CCD camera
- Axis pan/tilt/zoom camera

These components have been packaged into exchangeable modules: 2 computer modules, the GPS module, the data link module (wireless Ethernet, wireless serial, Ethernet switch), and the IMU module. These modules are placed in a vibration-isolated rack below the main body of the helicopter, shown in Figure 2. Each module has its own self-contained power regulation, air-cooling, and Electro-Magnetic Interference (EMI) shielding. There is also a sonar/magnetometer assembly at the tail, a power distribution system including circuit breakers near the module rack, and mounting points for camera systems and other components under the nose.

The power distribution system utilizes the onboard generator, which outputs 12V DC. It includes a hot-swappable connection to use external power. Each component has a dedicated individual circuit breaker.

Wiring external to the modules consists of RS-232 Serial, Ethernet, and 12V DC only. Wiring is routed to one side of the module rack, Figure 3. The other side is kept free, Figure 2, and available for temporary hookups (e.g. Ethernet), status LEDs, and switches. The complete wiring diagram is shown in Figure 4, including a typical configuration of RS-232, Ethernet, and power wiring. Note the compartmentalization in modules, dedicated power regulation within each module, and the interface to the YACS via multiple serial lines.



Figure 2- Vibration isolated avionics module rack, from left to right: data link module, GPS module, computer module #1 (flight computer), computer module #2 (auxiliary)



Figure 3 – Wiring on “back” side of module rack, note that in this particular picture computer module #2 on the left is empty

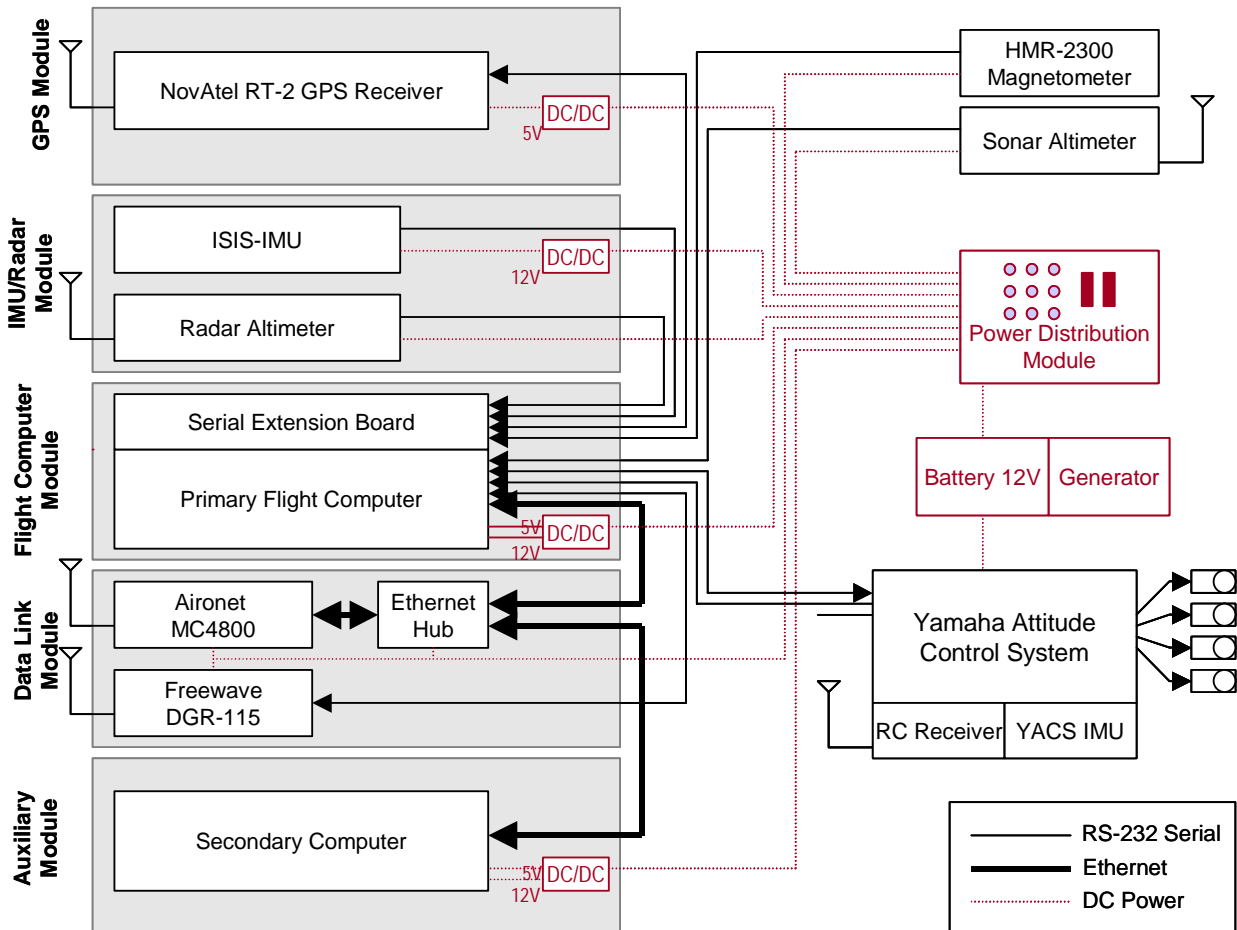


Figure 4: GTMax wiring diagram, including separation into modules, each with their own power regulation, air-cooling, and EMI shielding

The operating systems utilized for typical onboard (Flight) software are: VxWorks, QNX, Linux, or a combination. Operating system independence is maintained to maximize the ability to support varied research programs. The operating system independence is accomplished by extensive use of ANSI C/C++ (and the OpenGL API for graphics used in simulations and graphical user interfaces). No special compilers are utilized. Normally Microsoft Visual Studio is used for Windows and the GNU c-compiler is used for Linux and QNX.

The onboard software runs on the primary flight computer and/or secondary computer. The Ground Control Station (GCS) software runs on the ground, normally on one or more laptop computers, and is used to for human operators to interact with the onboard systems. Simulation-specific software is that not used in the flight configuration. All of the above software is included in the GCS or simulation-tool builds. Only the onboard software is included in a primary flight computer or secondary computer build.

The baseline navigation system running on the primary flight computer is a 17 state Extended Kalman Filter. The states include: vehicle position, velocity, attitude (quaternion), accelerometer biases, gyro biases, and terrain height error. The system is all-attitude capable and updates at 100 Hz<sup>10</sup>.

The baseline flight controller is an adaptive neural network trajectory following controller with 18 neural network inputs, 5 hidden layer neurons, and 7 outputs for each of the 7 degrees of freedom<sup>11,12</sup>. The 7 degrees of freedom include the usual 6 rigid-body degrees of freedom plus a degree of freedom for rotor RPM. The controller can also be configured as a conventional inverting controller. The baseline flight controller and navigation system, which coupled with the simple baseline trajectory generator, is capable of automatic takeoff, landing, hover, flight up to the maximum attainable by the helicopter (around 85 feet/sec) and aggressive maneuvering, discussed further in the results section below.

#### Data Communication

Generic and highly capable data communication software has been developed to support a large number of potential flight and simulator test configurations. First, these routines supports serial data reading and writing as necessary for the Commercial Off The Shelf (COTS) sensors and other custom components used.

These same routines can also be used to re-route any data through Ethernet or as memory within a single executable. These data routings can be modified in real-time, by software switch. It should be noted that

almost all operating system specific software is limited to these routines.

These routines are used to: interface with all sensors, the wireless serial data link, and to repeat all wireless serial data over the wireless Ethernet (for redundancy). Also, any data received over a link can be stored to a binary file. This recorded data can then be played back to stimulate selected components. All data received from the helicopter over either data link is stored in this manner during flight.

#### Simulation Tools

Early in the GTMax system design, the top-level simulation requirements to support the development and operation of an experimental UAV were identified as:

1. Test all custom developed software and guidance, navigation, and control algorithms in a rigorous manner (more extensively than practical or safe in a flight test setting)
2. Test onboard computer hardware, operating system implementation, and software execution in real-time
3. Rehearse all procedures and flight test plans
4. Visualize recorded flight test data
5. Reconstruction of flight test events after-the-fact (e.g., incident reconstruction)
6. Can be utilized at the flight test location
7. Flight test data validation

From these, the following component requirements were derived:

1. Models of the sensors, aircraft, and aircraft interfaces – down to the level of binary serial data (i.e., packets) with time delays
2. Injection of model error and environmental disturbances, of those expected to be experienced in flight and worse
3. Flexible scene generation capability
4. Can operate effectively using a single conventional laptop computer
5. Reconfigurable data communication routines discussed above in the context of flight software

The simulator tools that have been developed normally run on a high-end personal computers or laptops that uses the Windows 2000/NT operating system or Linux, and is written primarily in C/C++. It

includes an aircraft model, the aircraft interface model, and sensor models. The aircraft model has six rigid-body degrees of freedom plus engine, fuel, and rotor dynamics. The helicopter interface model simulates the servo interface unit functionality and RS-232 serial interface. The sensor models include errors, mounting location and orientation, time delays, and digital interfaces. The scene generator is a real-time 3-D graphics window, Figure 5, showing the aircraft and the terrain, and has additional functionality to aid in data visualization or use in the Ground Control Station<sup>13</sup>.

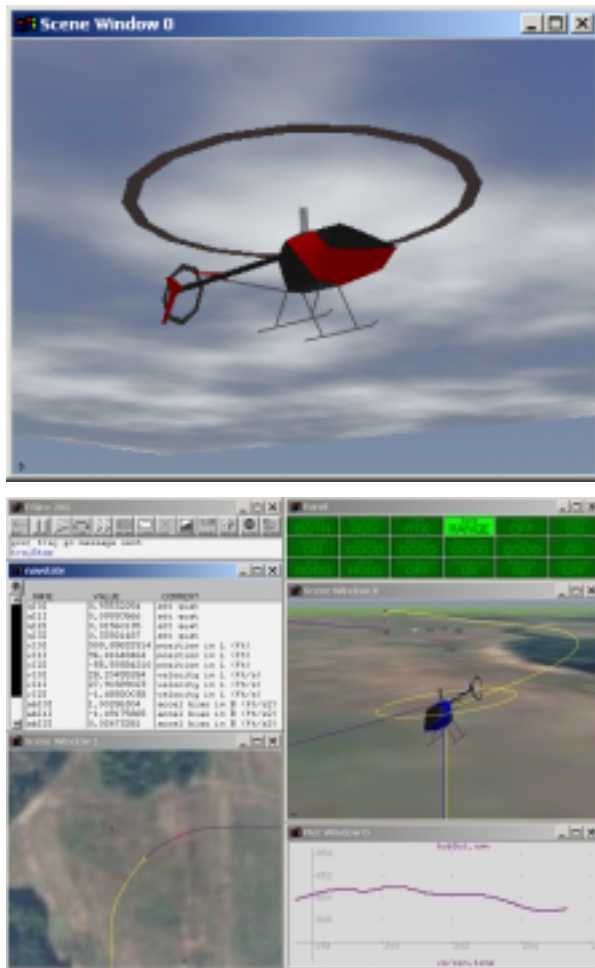


Figure 5 – Simulator scene generator (above) and its use in the Ground Control Station (GCS) interface (below)

The basic simulation tools allow for real-time display of all onboard data, including plotting and logging. It also allows one to modify any data. The simulator can run in real time or in a batch mode (faster than real time).

To utilize the common-form of the Software-In-The-Loop (SITL) simulation configuration, Figure 6, the un-compiled software source code, which normally runs on the onboard computer, is compiled into the simulation tool itself, allowing this software to be tested on the simulation host computer. This allows the flight software to be tested without the need to tie-up any flight hardware. Once any modification has been tested with the SITL configuration, any required Hardware-In-The-Loop (HITL) simulation configurations are used. The configurations required depend on what is being tested, but a test of a change to the primary flight computer guidance, navigation, and control system is shown in Figure 7. For this HITL simulation, the onboard computer is plugged into a simulation-host computer. Here, the hardware under test is the onboard computer(s), servos, along with all software that executes on the computer(s). The sensor and helicopter interface models provide the proper interfaces to the onboard computer, so the onboard computer configuration is identical to that used in a flight test. This HITL simulation configuration is used to test all guidance, navigation, and control algorithms software and as much of the hardware as practical, in real-time.

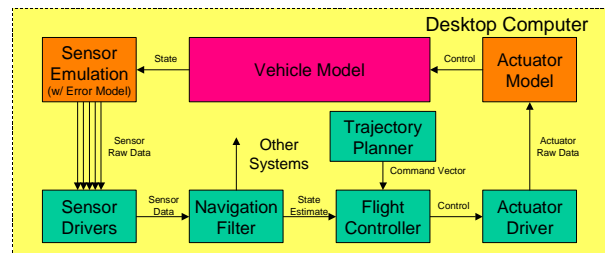


Figure 6 – Typical Software in the Loop structure

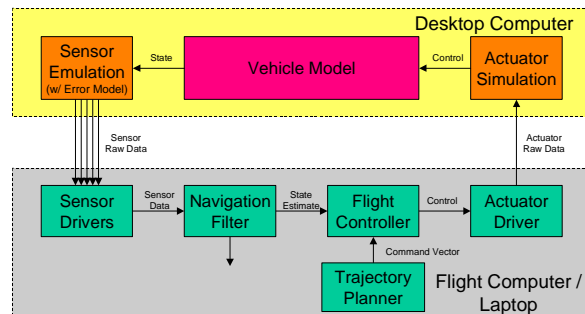


Figure 7 – Hardware in the Loop structure for testing changes to the primary flight computer software or hardware

## RESULTS

A number of research flight tests have been conducted on the GTMax, including the development of the baseline guidance, navigation, and control algorithms, SEC program tests, and the 2002 Aerial Robotics Competition. These results are summarized here. Video of many of these tests can be found on the web at [controls.ae.gatech.edu/uavrf](http://controls.ae.gatech.edu/uavrf).

### Neural Network Adaptive Flight Control

A Neural Network (NN) adaptive flight control system that tracks desired helicopter trajectories has been tested extensively on the GTMax<sup>11,12</sup>. The system includes the ability to handle large model errors. For these tests, a simple linear model corresponding to the hover flight condition is the only model information provided *a priori* to the system. Online training of the neural network is relied upon to correct for the resulting model error over the entire speed flight envelope of the helicopter. The system also inherently accounts for actuator position and rate saturation as well as time delay.

A typical result is shown in Figure 8 for a “racetrack” pattern viewed from above flown at 40 feet/second. This trajectory starts and finishes in hover and includes normal turns at each of four waypoints. The maximum distance between commanded and estimated position is approximately 10 feet, with average errors much lower. Figure 9 illustrates a typical speed trial, in this case reaching a maximum ground speed of approximately 95 feet/sec. The first leg is upwind, the second is downwind. On the upwind leg the speed is limited by saturating collective pitch. Rotor RPM is held constant throughout at 850. Figure 10 illustrates a typical aggressive maneuver, in this case a reversal of course, starting and finishing at 50 feet/second. The maneuver takes approximately 6 seconds, and allows the helicopter to come to a stop in approximately 75 feet.

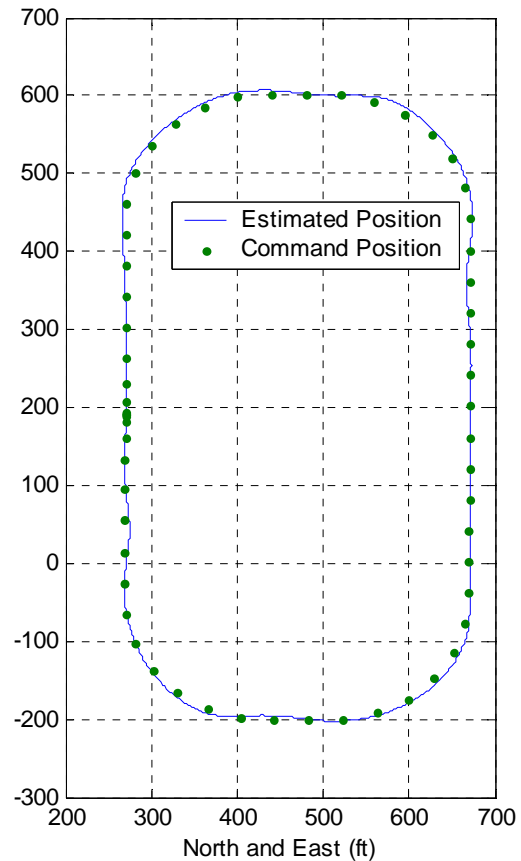


Figure 8 – Racetrack pattern flown at 40 feet/second starting and finishing in hover, commanded position plotted once per second

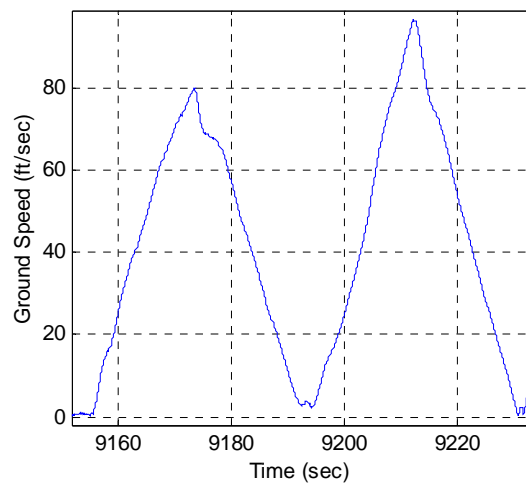


Figure 9 – Speed trial, first upwind (left) then downwind (right); speed limited by saturating collective upwind

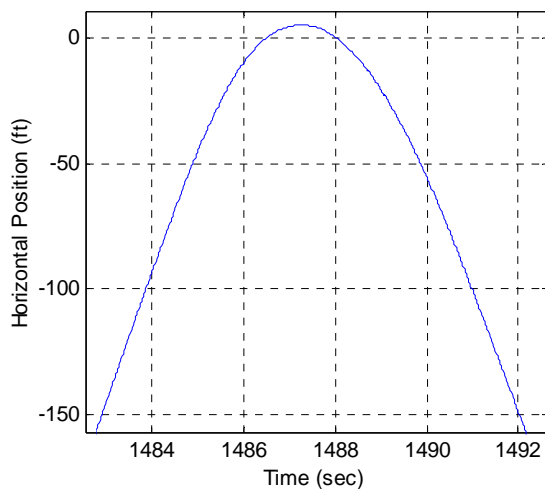
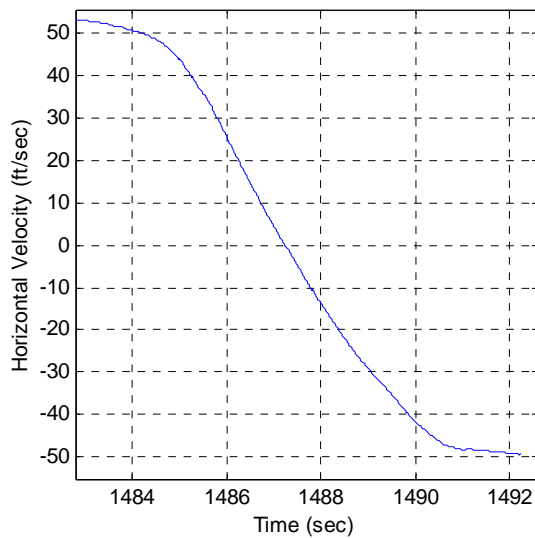


Figure 10 – Aggressive turn enabling stop and direction reversal in approximately 75 ft from 50 ft/sec; (top) shows velocity (bottom) shows position

### Open Control Platform for Flight Control Software

The Open Control Platform (OCP) was flight testing performing low-level flight control functions for the GTMax, including the reconfiguration of flight control software while the helicopter was in automatic hover. The architecture used is illustrated in Figure 11, where the baseline software modules were configured as OCP software components, and two copies of the controller component were developed. The first used the nominal adaptive flight controller; the second was configured as a conventional inverting controller. The OCP was able to swap between these two controller modules in real time, even with components running at 50 and 100 Hz.

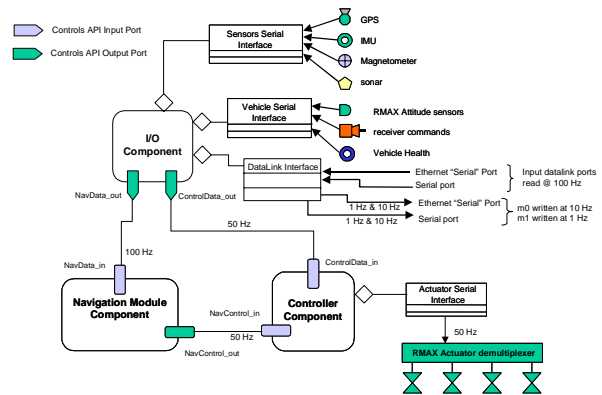


Figure 11 – OCP Implementation of GTMax Guidance, Navigation, Flight Control, and Communication

### Aerial Robotics Competition

In August 2002, the GTMax system was used to compete in the AUVSI International Aerial Robotics Competition. For this competition, a UAV system must automatically locate a specific building in a prescribed search area, and then identify an opening into the building. This must be done without any human assistance during a mission attempt. A camera and frame grabber were added to the basic GTMax, and the 2<sup>nd</sup> computer was configured as an image processing subsystem running the Linux operating system and utilizing the OpenCV library. Mapping and flight planning software components were added to the primary flight computer.

Due to radio frequency interference at the contest site, multiple attempts had to be aborted due to loss of GPS lock. However, the three buildings in the search area were found on three different attempts. On one attempt, the system was able to descend down to 50 feet of altitude and circle each of the three buildings looking for the identification marking and openings into the building before a GPS failure, shown in Figure 12. On this attempt, the correct building was identified, but the opening automatically selected was from a neighboring building.



Figure 12 – GTMax automatically locates buildings in search area during 2002 International Aerial Robotics Competition; here it is automatically looking for openings into the buildings

#### Automatic Takeoff and Landing

Automatic takeoffs and landings have been performed using the Neural Network adaptive flight control system described previously<sup>11,12</sup>. The navigation system determines if the helicopter is on the ground by comparing the altitude above ground level (AGL) with a preselected value. That is, if the helicopter is within a few inches of the ground it acts as though it is on the ground. Commanded position is slaved to the current position when the helicopter is declared on the ground. Also, internal states of the flight control system are frozen. Depending on whether the helicopter is attempting a takeoff or a landing/hold, the rotor RPM is ramped either up to flight speed or down to idle respectively. Also depending on objective, the collective pitch is ramped up or down at a constant rate. Takeoffs are performed by ramping rotor RPM and collective until the helicopter is detected airborne, at which point the trajectory generator produces a smooth climb trajectory. Landings end with a slow vertical descent command until ground contact is detected and rotor RPM and collective pitch are reduced to an idle state.

A plot of the first automatic takeoff of the GTMax is shown in Figure 13, where a smooth climb to 30 feet of altitude and a hover were specified. The first automatic landing and the GTMax is shown in Figure 14, where in this case a slow descent of 0.5 feet per second is used until ground contact is detected.

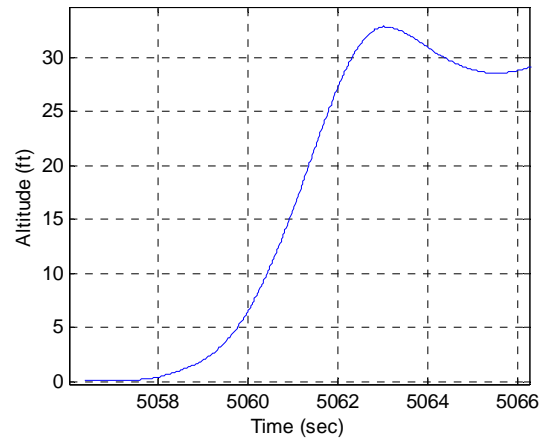


Figure 13 – Automatic takeoff and smooth climb to a hover at 30 feet of altitude

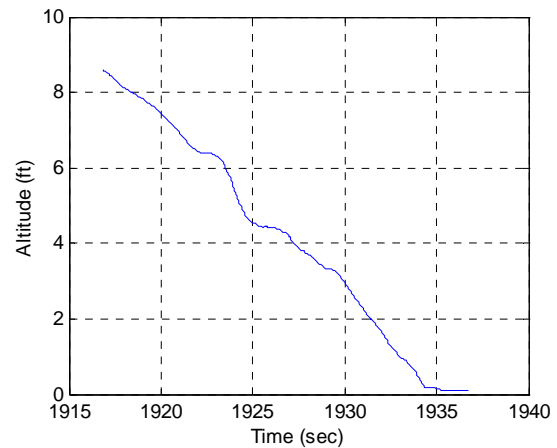


Figure 14 – Automatic landing, after a smooth descent at 0.5 feet/second

#### Automatic Flight Envelope Protection

An automatic flight envelope protection system that utilizes an online trained neural network to predict and then avoid flight envelope limits. Flight tests conducted to date include avoidance of a rotor stall prediction parameter (Erits factor, in units of speed), set artificially conservative to facilitate safe testing<sup>14</sup>.

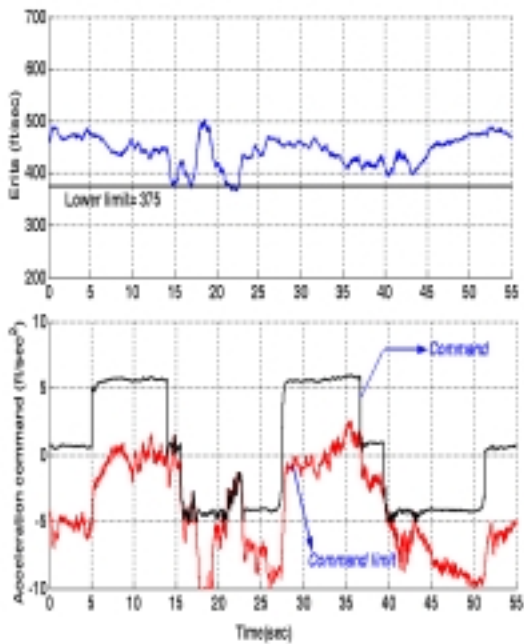


Figure 15 – Eris factor limit (a prediction of blade stall) is avoided by modifying the trajectory acceleration command (velocity and position command are also modified accordingly)

#### Fault Tolerant Control

The scenario of a stuck collective pitch actuator was simulated in flight by limiting the deflection of swash plate actuators in such a way to prevent changes in collective pitch. A fault tolerant control module was developed, running with the OCP on the 2<sup>nd</sup> flight computer, generated a rotor RPM command that allows the existing flight controller to continue to function, albeit at reduced capability. This capability could be enough to safely recover the vehicle in such a scenario. Further details will be published in a subsequent paper. The range of acceptable rotor RPM command was experimentally determined to be 700 to 950, all utilizing the baseline adaptive flight controller without modification (normally rotor RPM is 850). Note that flight at 700 RPM required saturated collective pitch in order to hover.

A typical flight test result is illustrated in Figure 16, where up and down step responses of 10 feet were performed in between hover segments. Altitude hold performance is significantly worse without collective pitch, but still effective. No changes were made to the other axes of flight control, but performance was not degraded significantly.

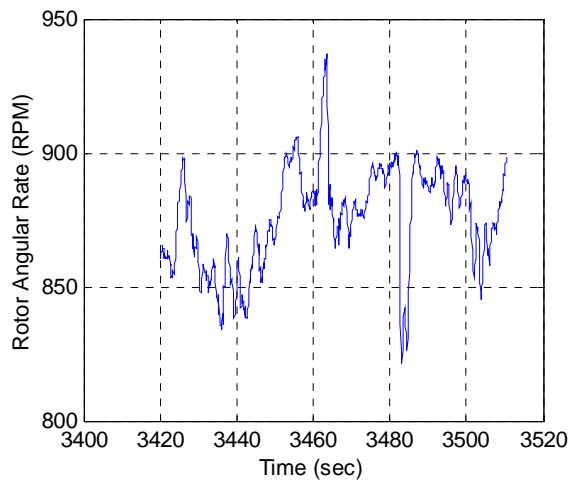
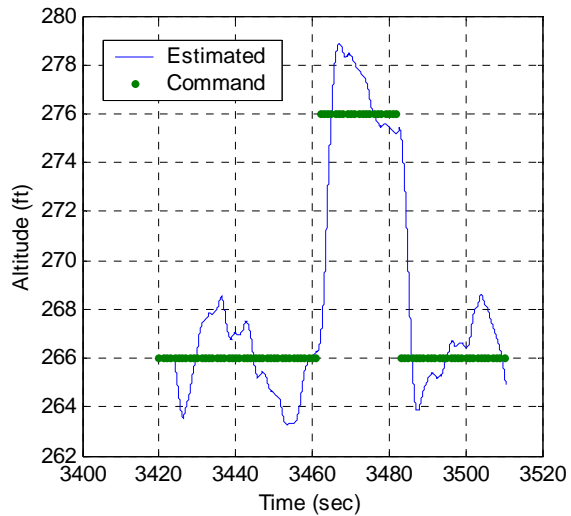


Figure 16 – Utilizing rotor RPM to accommodate a simulated stuck collective actuator, step responses; (top) altitude and altitude command, (bottom) measured rotor RPM; note transient in RPM during climb and descent

#### CONCLUSIONS

The extensive use of a variety of simulation configurations has been of considerable benefit for the recent development and operation of the GTMax UAV, and for its use in research and the International Aerial Robotics Competition. The key features of a flexible data communication system, models for all hardware components, and a simulation software infrastructure enable these configurations. The benefits have included increased safety, effective participation of a large number of researchers, the detection of errors before flight testing, and the effective use of flight test data.

The use of a modular avionics architecture and the use of a vehicle with a relatively large payload capacity allows the GTMax to be configured quickly for a variety of experiments. For example, it allowed the vehicle to be reconfigured for the aerial robotics competition relatively easily by adding a camera and video server. The capacity also allows efficient use of flight test time, allowing multiple unrelated flight test points during a single flight.

#### ACKNOWLEDGEMENTS

The authors would like to acknowledge some of the many other contributors to this work: Henrik Christophersen, Scott Clements, J. Eric Corban, Cameron Craddock, Graham Drozeski, Manuj Dhingra, Joerg Dittrich, Richard Guily, Luis Gutierrez, Murat Guler, Jincheol Ha, Bonnie Heck, Jeong Hur, Aaron Kahn, Suresh Kannan, Eric Martens, Brian Mendel, Sumit Mishra, Wayne Pickell, J.V.R. Prasad, Alison Proctor, Suraj Unnikrishnan, George Vachtsevanos, Linda Wills, and Ilkay Yavrucuk. This work is supported by the DARPA Software Enabled Control (SEC) Program under contracts #33615-98-C-1341 and #33615-99-C-1500.

#### REFERENCES

<sup>1</sup>Christophersen, H., Hart, M., Dhingra, M., Johnson, E., and Guily, R., "Development of an Autonomous Aerial Reconnaissance System at Georgia Tech," *Proceedings of the Association for Unmanned Vehicle Systems International Unmanned Systems Symposium & Exhibition*, 2002.

<sup>2</sup>Brinker, J., and Wise, K. "Flight Testing of a Reconfigurable Flight Control Law on the X-36 Tailless Fighter Aircraft." *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2000.

<sup>3</sup>Idan, M., Johnson, M., and Calise, A. "A Hierarchical Approach to Adaptive Control for Improved Flight Safety." *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 6, 2002.

<sup>4</sup>Johnson, E., Calise, A., El-Shirbiny, H., and Rysdyk, R. "Feedback Linearization with Neural Network Augmentation Applied to X-33 Attitude Control." *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2000.

<sup>5</sup>Corban, J., Calise, A., Prasad, J.V.R., Hur, J., and Kim, N. "Flight Evaluation of Adaptive High-Bandwidth Control Methods for Unmanned Helicopters," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2002.

<sup>6</sup>O. Shakernia, C. S. Sharp, R. Vidal, D. Shim, Y. Ma, and S. S. Sastry, "A Vision System for Landing an

Unmanned Aerial Vehicle," *IEEE Int. Conference Robotics and Automation*, 2002.

<sup>7</sup>M. La Civita, G. Papageorgiou, W. C. Messner, T. Kanade, "Design and Flight Testing of a High-Bandwidth H-infinity Loop Shaping Controller for a Robotic Helicopter," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2002.

<sup>8</sup>Nakanishi, H., Inoue, K., and Sato, A. "Development of Command and Control Systems for UAVs," *Proceedings of American Helicopter Society 57th Annual Forum*, 2001.

<sup>9</sup>Wills, L., Kannan, S., Sander, S., Guler, M., Heck, B., Prasad, J.V.R., Schrage, D., and Vachtsevanos, G., "An Open Platform for Reconfigurable Control," *IEEE Control Systems Magazine*, pp. 49-64, June 2001.

<sup>10</sup>Dittrich, J., and Johnson, E., "Multi-Sensor Navigation System for an Autonomous Helicopter," *Digital Avionics Systems Conference*, 2002.

<sup>11</sup>Johnson, E. and Kannan, S., "Adaptive Flight Control for an Autonomous Unmanned Helicopter," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2002.

<sup>12</sup>Kannan, S. and Johnson, E., "Adaptive Trajectory-based Flight Control for Autonomous Helicopters," *Proceedings of the 21st Digital Avionics Systems Conference*, 2002.

<sup>13</sup>Johnson, E., and Mishra, S., "Flight Simulation for the Development of an Experimental UAV," *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, 2002.

<sup>14</sup>Yavrucuk, I., Prasad, J.V.R., Calise, A., and Unnikrishnan, S., "Adaptive Limit Control Margin Prediction and Avoidance," *58th AHS Annual Forum*, 2002.