# A Low Cost Small UAV Flight Research Facility

Austin M. Murch, Yew Chai Paw, Rohit Pandita, Zhefeng Li, and Gary J. Balas

**Abstract.** This paper presents an overview of the low-cost, open source small Unmanned Aerial Vehicle (UAV) flight research facility at the University of Minnesota. A detailed description of the facility, its components, and its capabilities is presented, as well as applications of the UAV platforms to research, education, and collaboration. Flight results from controls research is presented, followed by a brief summary of current research and future directions.

## 1 Introduction

Unmanned Aerial Vehicles (UAVs) are used worldwide today for a broad range of military, civil, and research applications. There continues to be a growing demand for reliable and low cost UAV systems. This is especially true for small to miniature-sized UAV systems (less than 2 meter wing span) where the majority of systems are still deployed as prototypes due to demand and lack of reliability. Improvements in the modeling, testing and flight control for these vehicles would help to increase their reliability and performance of small UAVs during operations. The traditional approach used in the development cycle [1, 2] for manned aircraft is time consuming and resource intensive. Applying the same techniques to the small UAVs is not realistic.

The UAV Research Group in the Department of Aerospace Engineering and Mechanics (AEM) at the Univeristy of Minnesota is focused on development and implementation of a low-cost, open source small Unmanned Aerial Vehicle (UAV) flight research facility. The goal of this facility is to support research activities within the department including control, navigation and guidance algorithms, embedded fault detection methods, and system identification tools. The system is built mostly out of commercial-off-the-shelf (COTS) components to minimize the overall material and development costs. In addition, the entire architecture is open and available

Austin M. Murch · Yew Chai Paw · Rohit Pandita · Zhefeng Li · Gary J. Balas
University of Minnesota, Minneapolis, MN 55455
e-mail: {murch,paw,pandita,zhefeng,balas}@aem.umn.edu

to any researchers or organizations who wish to collaborate on the development or application of the UAV capabilities. (http://www.aem.umn.edu/∼uav/) In addition to the researchers from AEM department, the UAV Research Group is collaborating with researchers at the Budapest University of Technology and Economics in Hungary and the University of Sannio at Benevento, Italy.

## 2 UAV Testbeds

The UAV Research Group uses COTS R/C fixed wing aircraft modified to carry the necessary avionics and instrumentation payloads. Several different aircraft models have been used during the development process, but the primary test aircraft is the Ultra Stick 25e [3] (shown in Figure 1(a)). Two additional aircraft in use are an Ultra Stick 120 (shown in Figure 1(b)) which can handle significantly more payload than the Ultra Stick 25e, and the Mini Ultra Stick, the smallest version of the Ultra Stick family.


(a) Ultra Stick 25e (b) Ultra Stick 120 (FASER)

**Fig. 1** Ultra Stick UAV Testbeds

The Ultra Stick 120 (aka FASER; Free-flying Aircraft for Subscale Experimental Research) has a conventional horizontal and vertical tail with rudder and elevator control surfaces, respectively. The aircraft has a symmetric airfoil wing with aileron and flap control surfaces. All six control surfaces are actuated by Hitec HS-5625MG servos. The plane is propelled by a 1900W Actro 40-4 brushless electric motor with a Graupner 14 x 9.5 folding propeller. Power for the motor comes from two 5000mAh 5-cell lithium polymer batteries connected in series. The servos are powered by a separate 1350 mAh 3-cell lithium polymer battery. The main internal payload bay is located directly under the wing in the fuselage and measures approximately 35cm L x 10cm H x 10cm W; additional payloads may be accomodated in the aft fuselage or externally. The 120 class can carry approximately 2.5kg of payload.

The Ultra Stick 25e is an approximately 65% scale model of the Ultra Stick 120, with the same basic configuration. The UAV Research Group maintains three Ultra Stick 25e aircraft, named 'Odin', 'Loki', and 'Thor'. All six control surfaces are actuated by Hitec HS-225BB servos. The plane is propelled by a 600W E-Flite Power

**Table 1** Summary of Aircraft Parameters

| Parameter | Mini Ultra Stick | Ultra Stick 25e | FASER |
|---|---|---|---|
| Wing span | $0.985\,m$ | $1.27\,m$ | $1.92\,m$ |
| Wing chord | $0.21\,m$ | $0.3\,m$ | $0.43\,m$ |
| Length | $0.865\,m$ | $1.05\,m$ | $1.32\,m$ |
| Wing reference area | $0.21\,m^2$ | $0.32\,m^2$ | $0.769\,m^2$ |
| MTOW (Tested) | $N/A$ | $2.04\,kg$ | $9.07\,kg$ |
| Empty weight | $0.62\,kg$ | $1.50\,kg$ | $6.35\,kg$ |
| Endurance | $10-15\,min$ | $15-20\,min$ | $15-20\,min$ |
| Cruise speed | $10-15\,m/sec$ | $15-20\,m/sec$ | $20-30\,m/sec$ |

25 brushless electric motor with an APC 12 x 6 propeller. Power for the motor and servos comes from a 4200mAh 3-cell lithium polymer battery. The main internal payload bay is located directly under the wing in the fuselage and measures approximately 22cm L x 6cm H x 7.5cm W; additional payloads may be accomodated in the aft fuselage or externally. The 25e class can carry approximately 0.55kg of payload.

The Mini Ultra Stick is an approximately 50% scale model of the Ultra Stick 120, with the same basic configuration. The UAV Research Group currently uses this aircraft as a wind tunnel model, as the smaller size enable it to fit into the AEM department's low speed wind tunnel.

The specifications of the three Ultra Stick aircraft are given in Table 1. The Ultra Stick aircraft family is marketed as a trainer-level aerobatic aircraft, so it is relatively easy for a moderately skilled R/C pilot to fly, but is still capable of highly dynamic manuevering flight.

## 3 Onboard Avionics

The current architecture of the onboard avionics is shown in Figure 2 and Table 2 gives a listing of the individual components in the system. The flight computer uses a real-time operating system and flight software written in C. The flight computer handles data collection from each sensor, performs attitude and position estimation, executes flight control algorithms, stores relevant data, outputs PWM servo commands, and sends information to the ground control station via the data modem. A failsafe switching board is used to switch control of the aircraft between manual mode (human R/C pilot) and the flight computer. The hardware interface to the flight computer is handled via a custom-designed interface board.

### *3.1 Sensors*

The sensor suite for the UAVs is focused on measuring the aircraft state data needed for normal flight guidance, navigation, and control algorithms. Position, velocity,

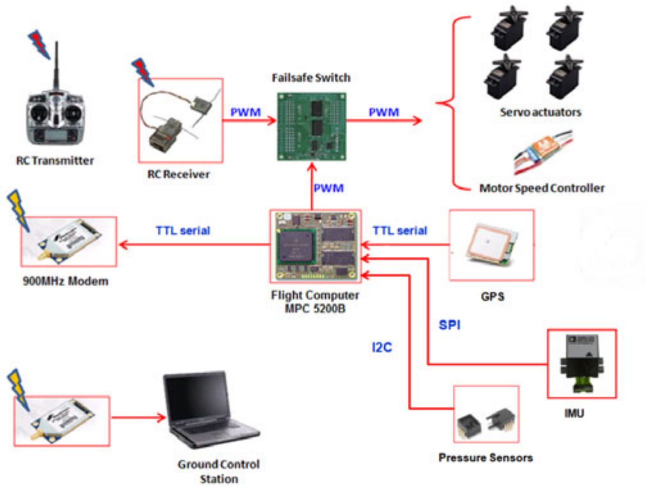



**Fig. 2** Onboard avionics architecture

**Table 2** Onboard avionics system components

| Component | Module | Cost |
|---|---|---|
| Flight computer | Phytec MPC5200B microcontroller | $300 |
| Interface board | AEM custom design | $250 |
| IMU | Analog Devices iSensor ADIS16405 | $800 |
| GPS | USGlobalSat EM-406A | $50 |
| Pressures | Honeywell ASDX | $60 |
| Data Telemetry | Free Wave MM2 900 Mhz modem | $375 |
| Failsafe Switch | AcroName RxMux | $300 |
| Manual Control | Spektrum DX-7 2.4 Ghz R/C system | $300 |

and accelerations in the aircraft body frame and navigation frame are the normal quantities of interest. In general, the minimum sensor suite to achieve these data with sufficient accuracy is the combination of an IMU, GPS, and pitot-static system. Prior work has utilized an integrated sensor suite (Crossbow Micronav) that combined all of these functions, but this product is no longer available or supported. An affordable replacement for the Micronav has been difficult to find, and as a result, individual sensor components have been selected and integrated into the system. While this process does incur development and testing overhead, it has the advantage of allowing the researcher to select optimal sensors for the given application, and also offers a simple path for upgrading sensor capabilities. The replacement sensor components are described as follows:

IMU sensor: The Analog Devices iSensor ADIS16405 is a small, low-cost, temperature compensated, tri-axial accelerometer, rate gyro, and magnetometer. Data is provided to the flight computer via SPI bus at 50Hz.

GPS module: The USGlobalSat EM-406A is a small 25mm square unit with an integrated antenna. The GPS circuitry is located directly underneath the patch antenna, allowing a very compact operation. The EM-406A uses the SiRF StarIII GPS engine and data is provided to the flight computer via TTL serial at 1Hz.

Pressure sensors: The Honeywell ASDX is a small IC-based, digital output, temperature-compensated pressure sensor available in differential and absolute versions. These units are used to measure total and static pressure to determine the airspeed and altitude of the aircraft, and are connected to a Pitot-static probe that protrudes forward of the right wing of the aircraft. Pressure data is provided to the flight computer via I2C bus at 50Hz.

## *3.2 Flight Computer*

The current flight computer is a phyCore MPC5200B-tiny 32-bit PowerPC microcontroller. Prior work has utilized an phyCore MPC55, but limitations in processing power motivated an upgrade. The MPC5200B has a clock frequency of 400 Mhz, 760MIPS of processing power, and performs floating point computation. Current flight software utilizes about 2% of the CPU capacity. It has a wide range of I/O capabilities to support communication with external devices in addition to onboard data storage capacity. Details on the specifications of the MPC5200B can be found on the Phytec website [4]. Data is stored onboard the MPC5200B in the 64MB SRAM and downloaded after the flight via an Ethernet connection to the ground station.

### 3.2.1 Flight Software

The onboard flight computer utilizes a real-time operating system (eCos) and flight software written in C. In addition to being open source and freely available, the eCos operating system provides a real-time kernel, can be configured to minimize the computing overhead required for the operating system, supports multi-threading, and is POSIX C compatible [5].

The flight software uses multiple prioritized threads running at different rates to perform the necessary tasks for flight. Prioritizing multiple threads helps to ensure the most important time critical tasks are performed first (e.g. data acquisition).

**Table 3** Flight software thread description

| Priority | Thread | Description | Frequency (Hz) |
|---|---|---|---|
| 1 | DAQ | data acquisition for each sensor component | 50 |
| 2 | AHRS | attitude determination using EKF | 50 |
| 3 | CLAW | flight control law and actuator commands | 25 |
| 4 | INS-GPS | INS/GPS navigation filtering algorithms | 10 |
| 5 | DATA | onboard data storage | 50 |
| 6 | TELE | packing and sending of telemetry data | 20 |

Currently, there are 6 threads, listed in Table 3, which describes their function, priority, and update rate.

The data acquisition, data storage, and telemetry threads are relatively stable and few changes are made to these software functions from flight to flight. Most research activity occurs in the AHRS, flight control law, and INS/GPS software functions.

## 4 Ground Control Station

The Ground Control Station (GCS) is used during flight testing to monitor the aircraft state and health status. It consists of a laptop computer running the GCS software connected via serial to a data modem. The GCS software is a Java-based program inspired by the Open Source Glass Cockpit Project. It is designed to give vital flight information in real time to observers in order to assess the flight performance and maintain situational awareness of the aircraft during the flight test. The GCS software includes a Heads-Up Display, a moving map showing the location of the aircraft, commanded actuator positions, and indicators to display flight control mode information. Figure 4 shows a screenshot of the GCS software.

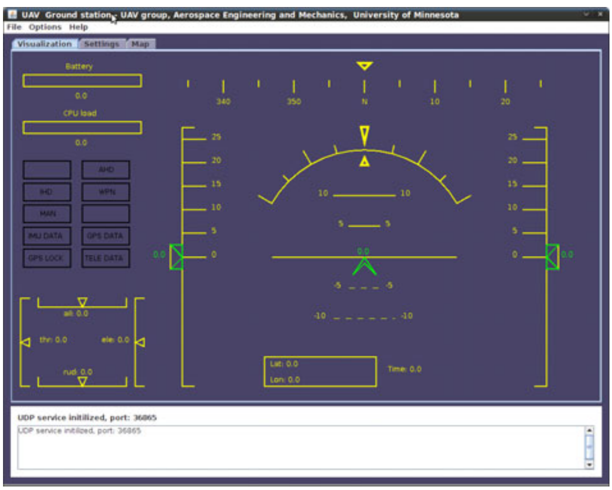



**Fig. 3** Ground control station

## 5 Simulation Testing

The use of simulation-based development and testing prior to actual flight testing reduces the total development time and helps ensure the algorithm under development is validated and bug-free, reducing costly debug time in the field and minimizing risk to the UAV platforms. The UAV Research Group maintains an integrated framework of three simulation environments used during the development process (Figure 4). The UAV simulation model is constructed in the Matlab/Simulink

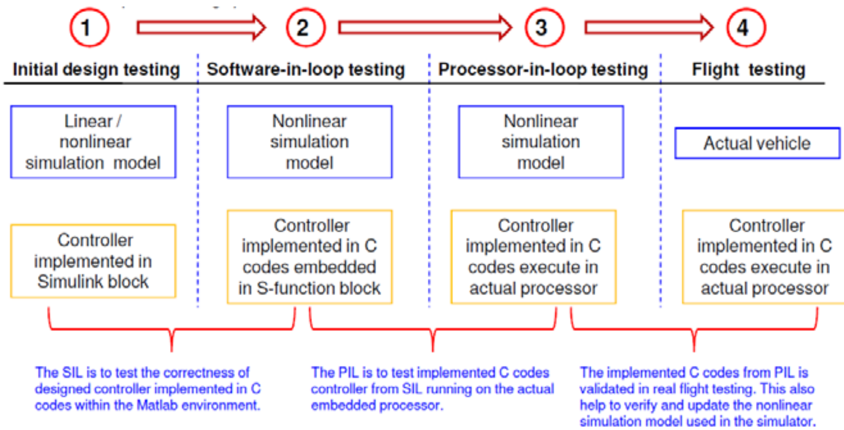



**Fig. 4** Application of simulation testing for flight control development

environment using the Aerospace Blockset [6]. All three simulations share the same nonlinear dynamics model.

## *5.1 Nonlinear Simulation*

The 6-DOF nonlinear simulation model uses the full nonlinear equations of motion, linear derivative aerodynamics, and table lookup propulsion models. Models of relevant aircraft subsystems such as actuators, motor, propeller, sensor dynamics, and noise are included. The environmental model includes a detailed model of Earth's atmosphere, gravity, magnetic field, wind, and turbulence. The aerodynamic derivatives were derived from first principles and empirical methods, and then updated using flight test data [7]. Bifilar swing tests were used to determine the moments of inertia, and wind tunnel testing was used to characterize the motor and propeller thrust, torque, and power.

## *5.2 Software-in-Loop Simulation*

The Software-in-the-Loop (SIL) simulation contains the nonlinear simulation with the C-code implementation of the control law in a Simulink S-function block. Only the flight control law is used at this stage; the remainder of the flight software is not included. This step is primarily used for debugging and to verify the C implementation of the algorithm under development matches the designer's expectations.

## *5.3 Processor-in-Loop Simulation*

The PIL simulation is an extension of SIL simulation that includes the MPC5200B flight computer into the simulation setup. Figure 5 shows the differences between

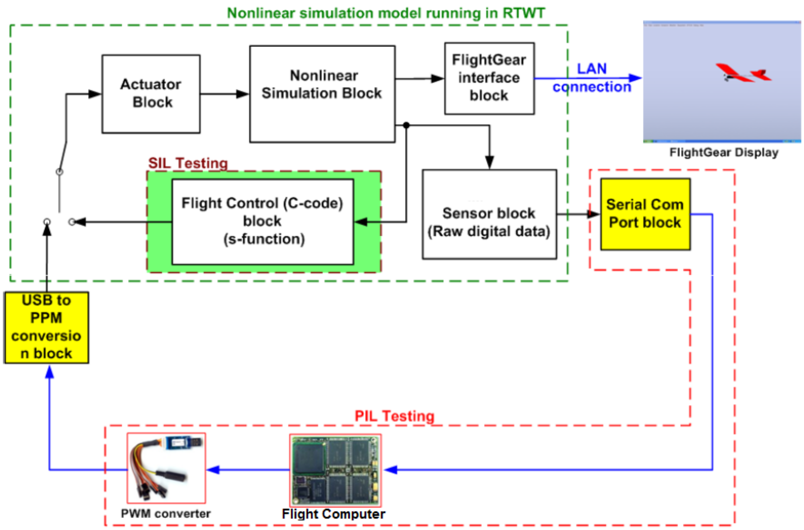



**Fig. 5** Block structure of SIL and PIL simulation

the SIL and PIL simulations. The Mathwork's Real-Time Windows Target toolbox [8] is used to ensure the simulation runs in real time on a Windows PC which is crucial for meaningful results when the real-time flight computer is included in the simulation loop. The PIL simulation also has an interface for a R/C pilot via a USB R/C-transmitter-style interface. Aircraft state data can be visualized on the GCS software or via FlightGear, a free open-source flight simulator [9]. The PIL simulation offers the following additional benefits to SIL simulation:

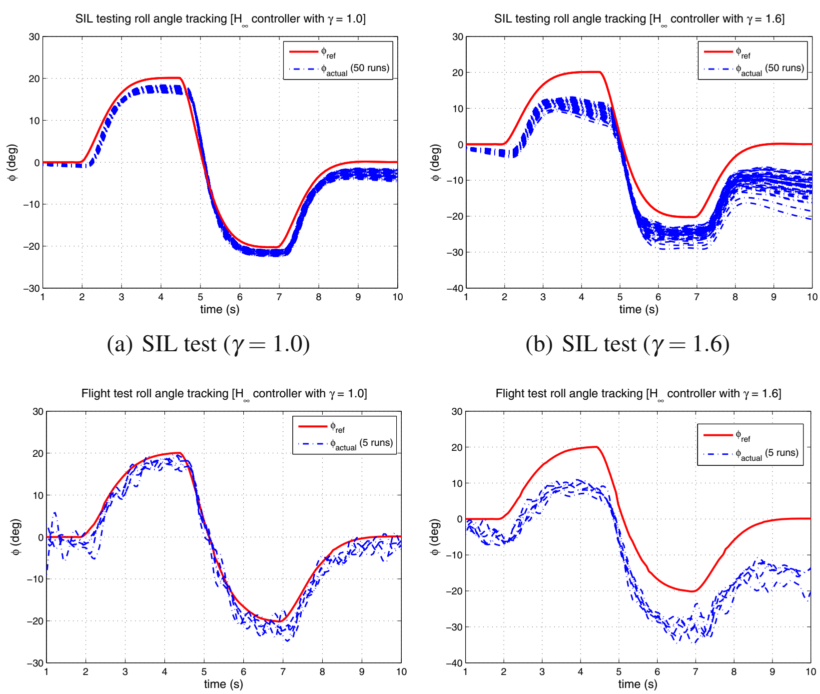
- Ability to test and identify controller implementation issues before the flight testing. This helps to determine the limitations on actual hardware and provides important information for controller redesign.
- Provides a real-time testing environment for synthesized controllers.
- Provides a testbed for integration and testing of hardware and software subcomponents at the system level.
- Provides an environment for the pilot and flight test engineers to prepare and understand the scope of the flight test and gain confidence in the overall system.

Beside testing, debugging and validating the control design and implementation, the PIL simulation is used for post-flight analysis in validating the simulation model that is recursively updated from flight test data. Once the simulation model has been sufficiently validated, it can be used to augment and substitute much of the flight testing which helps to reduce the risk and developmental costs of the system.

## 6 Flight Testing

Flight testing is the final stage of testing in the development cycle. Under the current concept of operations, a R/C pilot is the pilot in command and has authority over what has control of the aircraft (human vs. flight computer). The R/C pilot performs the takeoff and landing of the aircraft and transfers control to the flight computer once at an appropriate flight condition. A toggle switch on the R/C transmitter is used to transfer control of the aircraft to the flight computer via the RxMux fail-safe switch. At anytime during testing, the R/C pilot can take over control from the flight computer by toggling this switch on the R/C transmitter. Future plans include development of autonomous takeoff and landing capabilities, but the R/C pilot will remain as a safety backup.

Flight operations are contained to be within visual range of the R/C pilot and under 400 feet AGL altitude. Typical flight durations are 15-20 minutes, depending on the purpose of the flight. A normal deployment will last 3-4 hours and have 6-9 flights.



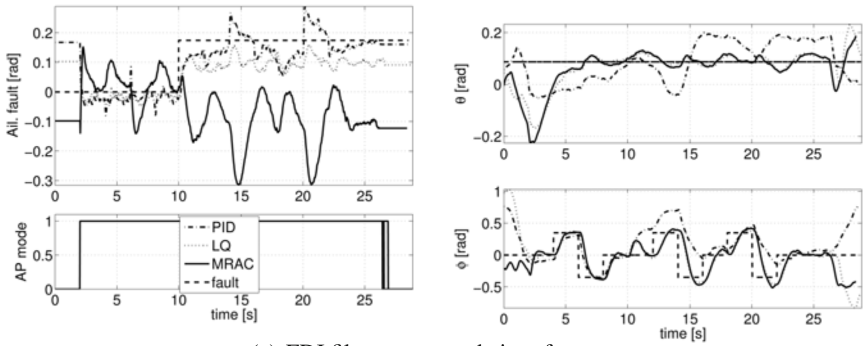

(a) SIL test ($\gamma = 1.0$)

(b) SIL test ($\gamma = 1.6$)

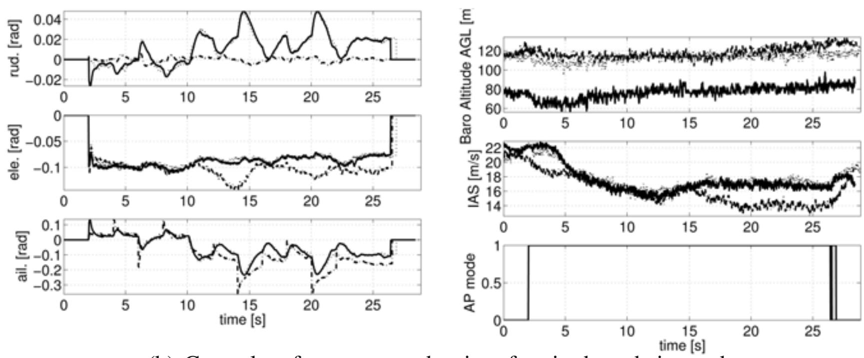
(c) Flight test ($\gamma = 1.0$)

(d) Flight test ($\gamma = 1.6$)

**Fig. 6** SIL and flight test result of $H_\infty$ controllers




(a) FDI filter output and aircraft states




(b) Control surface commands, aircraft attitude and airspeed

**Fig. 7** Experimental flight data analysis

Flight testing to date has primarily been focused on robust control law and fault detection filter research. Figure 6 shows a comparison of SIL simulation data and flight data for an $H_\infty$ control design [7]. This research effort was successful in using the entire UAV framework to develop, test, and fly a flight control law experiment. This effort also validated the integrated framework design approach and the concept of using a low-cost UAV platform for flight controls research. An acceptable match between the simulation and flight results was achieved despite the low fidelity simulation model and the relatively low quality sensors on the UAV platform.

The model-based fault detection and isolation (FDI) algorithms evaluated using the UMN flight test facility were based on robust $H_\infty$ filter designs. The effect of various closed-loop controllers on the FDI filter performance for an aileron fault were investigated. Three lateral-directional axis controllers were considered: a classical PID design, a LQ optimal multivariable design and a direct model-reference adaptive controller (MRAC). The experimental objective were to compare the robustness of the three controllers and the performance of the FDI filter to detect the aileron fault in the presences of the three controllers. Figures 7a and 7b show the flight test tracking performance of the three controllers. Recall that the controllers being investigated are associated with the lateral-directional axes of the aircraft. An identical pitch controller is used in all experiments to track 5 deg pitch command reference. The parameter *'AP mode'* indicates the status of FCC, i.e. whether the pilot or onboard flight program (OFP) is in control. The experimental results show the varied behavior attained with the robust filter for the different controllers. The benefit of having a flight test platform is the ability to validate theory and simulations on the test aircraft. In 2010, a total of 69 research flights were conducted.

## 7 Current Activities and Future Directions

Currently, the UAV Research Group is pursuing several research appliations for the UAV platforms and development framework.

- The National Science Foundation (NSF) has recently started a program focused on Cyber-Physical Systems (CPS) research [10]. As part of this research, the UAV platforms will play a key role in testing embedded fault detection algorithms including model based methods, software methods, and data-driven anomaly detection methods. These fault detection methods will also be applied to real-world industry problems in a collaborative effort with local industry to a production UAV platform and air data sensors.
- Precision landing of small UAVs is a significant challenge given the low quality sensors typically used. Many operational losses of UAVs result from the inability of the UAV to return reliably to a small protected location. The UAV simulation and platform will be used to develop precision landing algorithms, sensor requirements, and test results in an effort to improve the operational efficiency and loss rate of small UAVs.
- Extensive wind tunnel data is expensive to collect and is rarely openly available. Researchers at the NASA Langley Research Center performed significant static wind tunnel tests of an Ultra Stick 120 as part of the FASER development effort [11, 12]. This data is being implemented into the UAV simulation and will be openly available for research, education, and collaboration. This data set includes control surface and thrust effects over an angle of attack range of -5 to 40 degrees and an angle of sideslip range of +/- 30 degrees. As mentioned previously, the UAV Research Group operates an Ultra Stick 120 that was donated by NASA LaRC. This aircraft is instrumented with two angle of attack/sideslip vane sensors on wingtip booms, which expands the possible research applications of this platform to include aerodynamic model identification, gust alleviation, and high angle of attack/post-stall flight.
- The AEM department has developed a "Design, Build, Simulate, Test and Fly a UAV" course focused on the use of rapid prototyping software tools for vehicle modeling, guidance, navigation, and flight control, real-time implementation, hardware-in-the-loop simulation and flight tests. The UAV platforms and simulation framework was used for the class. Students repeated the design process of the UAV, including component selection, simulation, flight controller design, testing and implementation. At the end of the class, student-designed controllers were flight tested on the UAV platform.

## 8 Conclusion

A low-cost, open source small UAV flight research facility has been developed at the University of Minnesota. The UAV Research Group is actively applying the UAV platform to flight controls, guidance, navigation, and fault detection research. The AEM department is integrating the UAV platforms into the educational curriculum, giving students a unique opportunity to work with real flight data and have access to a flight test capability. The total cost required to field the UAV aircraft, avionics, and sensors described in this paper is under 3,000USD, which is less than the cost of most COTS autopilot systems marketed at small UAVs!

## References


1. Brian, S., Lewis, F.L.: Aircraft Control and Simulation. Wiley Interscience, Hoboken (2003)
2. Pratt, R.W.: Flight Control Systems: Practical Issues in Design and Implementation. Institute of Engineering and Technology (2000)
3. E-Flite, Ultra stick 25e arf, http://www.e-fliterc.com/Products/Default.aspx?ProdID=EFL4025
4. Phytec. MPC5200B microcontroller datasheet, http://www.phytec.com/products/sbc/PowerPC/phyCORE-MPC5200B-tiny.html
5. eCos, Homepage, http://ecos.sourceware.org/
6. I. The MathWorks, Matlab and simulink, http://www.mathworks.com
7. Paw, Y.C.: Synthesis and validation of flight control for UAV. Ph.D. thesis, University of Minnesota (2009)
8. Matlab, Simulink, Real-Time Windows Target 3 User's Guide. The MathWorks Inc. (2008)
9. FlightGear, Homepage, http://www.flightgear.org/
10. National Science Foundation, Cyber-physical systems (CPS) program, http://www.nsf.gov/pubs/2010/nsf10515/nsf10515.htm
11. Owens, D.B., Cox, D.E., Morelli, E.A.: Development of a low-cost sub-scale aircraft for flight research: The FASER project. In: 25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference (2006-3306)
12. Morelli, E.A., De Loach, R.: Wind tunnel database development using modern experiment design and multivariate orthogonal functions. In: 41st AIAA Aerospace Sciences Meeting and Exhibit (2003-0653)