AIAA 2001-4059

USE OF FLIGHT SIMULATION TO COMPLEMENT FLIGHT TESTING OF LOW-COST UAVS

Eric N. Johnson* and Sébastien Fontaine†
School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150

Abstract

The use of flight simulation tools to reduce the risk and required amount of flight testing for complex aerospace systems is a well recognized benefit of these tools. However, some special challenges arise when one attempts to extrapolate these benefits to very low-cost Uninhabited Aerial Vehicle (UAV) flight testing. These types of vehicles are characterized by a lack of payload capacity (and therefore limited in their capacity for additional flight test instrumentation or telemetry), limited baseline capabilities (processing, instrumentation, and telemetry), and by a lower marginal cost of added flight tests (less fuel, personnel, etc. is required per flight test, and a lower cost associated with a failure). In this paper, flight simulation architectures for system design and integration of low-cost UAVs are explored, and the development of a simulation tool for a low cost UAV described.

Introduction

It is essential to develop highly capable simulation tools to develop and test modern digital avionics systems. The sheer complexity of these systems necessitates the efficiency and control afforded by a simulation tool.

Today, there is also considerable interest in very small Uninhabited Aerial Vehicles (UAVs) and micro air vehicles, which have much of the guidance and control functionality typical of more traditional UAV designs, but are characterized by little redundancy, minimal instrumentation, and a minimal sensor payload. Traditional information such as air data or inertial information may not be available at all. This is a direct result of significant weight, power, and cost constraints associated with this class of vehicles, and is enabled by methods of estimating system parameters based on given sensor data and a model of vehicle behavior, such as[1]. By the same token, there is normally little or no margin to temporarily add instrumentation or telemetry to support a flight test program. This clearly can limit ones ability to extract all possible information during a flight test program.

There are, however, important benefits to these vehicles when it comes to flight testing. First, they typically have a much smaller marginal cost associated with additional flight testing. Second, risk of personal injury and the cost associated with a loss of a vehicle is normally smaller – although managing these risks still remains a high priority.

Here, the implementation for a specific low cost UAV design with associated simulation tools is described. The chosen fixed-wing UAV platform is significantly under-instrumented. This UAV is able to perform guidance and control functions by relying on the reconstruction of missing sensor signals and on a relatively stable open-loop aircraft. The purpose of the associated flight test program is to validate the system design and implementation, including its sensors and guidance, navigation, and control algorithms.

Description of a Low-Cost UAV

In this work, the low-cost UAV is a fixed wing aircraft equipped with a single engine, one commonly used to by novice hobby remote control airplane operators, shown in figure 1. The aircraft and engine cost approximately $250 total. The avionics system is also low-cost, and consists of a single GPS receiver, a computer, power distribution, and servo actuators. The onboard computer and battery are mounted in enclosures on the sides of the aircraft, evident in figure 1. The DeLoreme Earthmate® GPS antenna and receiver in its small enclosure is mounted above the wing, also shown. Navigation and control algorithms execute on the onboard computer, an AR-9612B motherboard with a 386SX Intel processor, utilizing VxWorks as the operating system. A servo interface unit allows the computer to control the servo actuators. To support a flight testing program, this unit also allows a separate receiver to give it operating servo commands, so that a safety pilot can take over control of the aircraft if necessary with a conventional hobby transmitter and receiver.

* Assistant Professor, Member AIAA.
E-mail: Eric.Johnson@aerospace.gatech.edu

† Graduate Research Assistant, Member AIAA.

Figure 1 – Low cost aerial vehicle, with avionics mounted above wing (GPS receiver), and in boxes on the sides of the fuselage (computer and battery)

A functional block diagram of the avionics used for this UAV is given in figure 2. GPS position and velocity information is used to update estimates of vehicle position, velocity, acceleration, bank, and local wind velocity. Guidance and flight control algorithms use this information to enable the vehicle to fly a prescribed flight plan utilizing aileron, elevator, and throttle control. Note that the system does not have a telemetry system, so flight data is recorded in onboard memory.

The motivation of this design is the development of minimal sensor configurations, to enable lower cost UAV applications. Also, this technology may serve as a potential backup configuration in many different applications.

Simulation Requirements

The top-level requirements to support a flight test program of the low-cost UAV described above were identified as:

1. Test all custom developed software and guidance, navigation, and control algorithms extensively (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 (i.e., incident reconstruction)
6. Can be utilized at the flight test location

From these, the following additional requirements were derived:

1. Models of the GPS sensor, aircraft, and servo interface
2. Injection of model error and environmental disturbances
3. Scene generation capability
4. Can operate effectively using a single laptop computer

Description of Simulation Tools

The simulator tool that has been developed normally runs on a high-end personal computer or laptop that uses the Windows 95/98/2000/NT operating system, and is written primarily in C. It is includes three principal models: the aircraft model, the servo interface model, and the GPS model. The aircraft model is a six-degree of freedom model, with aerodynamic data based approximations which are based largely on dimensions of the aircraft[2,3]. This data is used to develop lookup tables for aerodynamic coefficients, based on angle of attack. The servo interface model simulates the servo interface unit and servos. The GPS model, which is the only sensor this system to be tested utilizes, includes a model of GPS errors and latency. Flight and ground testing of the GPS unit was used to develop this model.

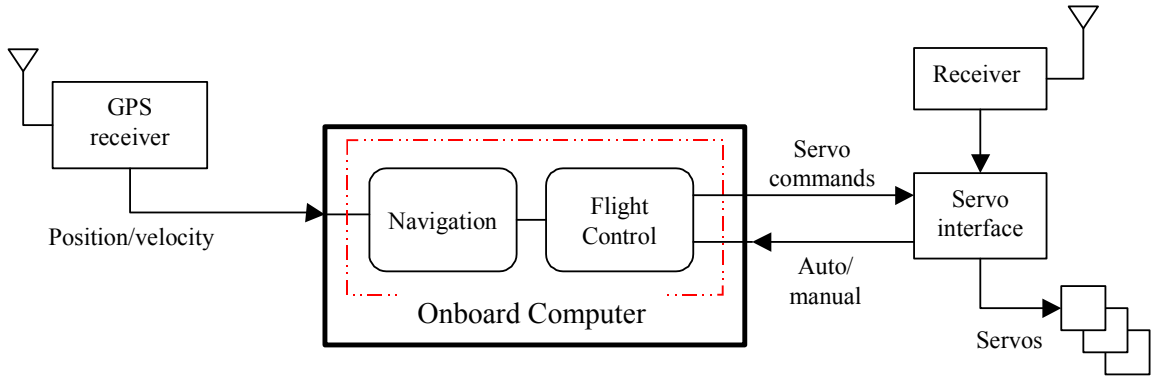

Figure 2 – Low cost guidance, navigation, and control system

The GPS model also emulates the RS-232 serial interface of the actual unit[4].

As shown in figure 3, the scene generator is a real-time 3D graphics window showing the aircraft, the Earth, terrain grid, and a runway. This allows the simulator output to be visualized. It is used for pilot training and mission rehearsal.

Figure 3 – Typical scene generator output

The scene generator allows different viewpoints, such as a ground observer, a "cockpit" view, and a chase plane view (shown in the figure). Salient variables such as speed, angle of attack, angle of sideslip, attitude, and speed can be displayed using a Heads-Up-Display (HUD)-like format. This provides an effective display of important variables and parameters of the simulator for users. Other simulation user interface components include 2-D plotting and a window for displaying and editing simulation data. These various windows are illustrated in figure 4.

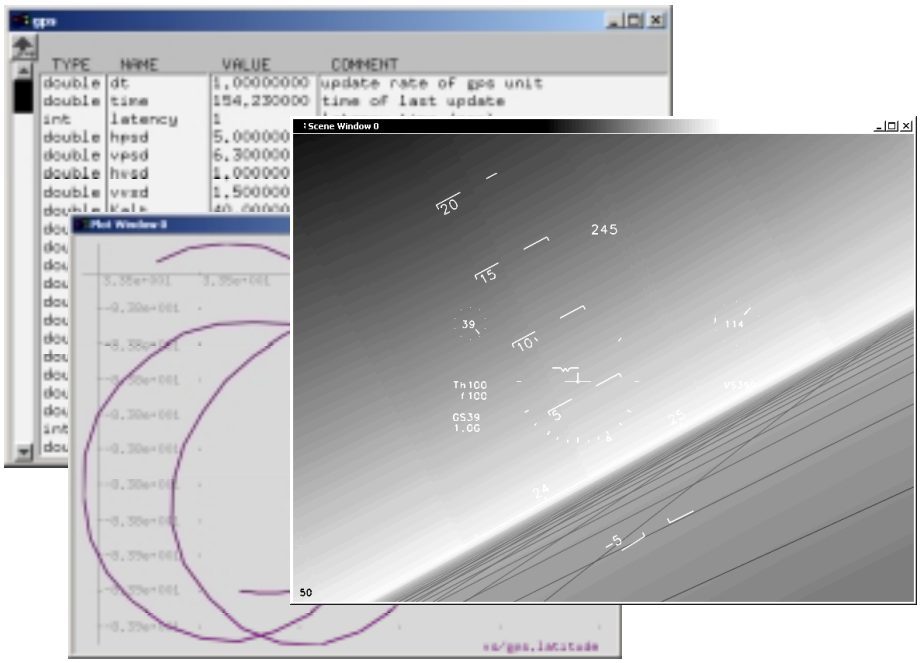

Figure 4 – Simulator User Interface Components

Software in the Loop

To utilize the Software-In-The-Loop (SITL) configuration, 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 the flight hardware, and was also used in selection of hardware. This configuration is illustrated in figure 5. This configuration is also used in the field to support flight testing, by operating on a high-end laptop computer, shown in figure 6. Once any modification has been tested with the SITL configuration, the Hardware-In-The-Loop (HITL) simulation configuration is used.

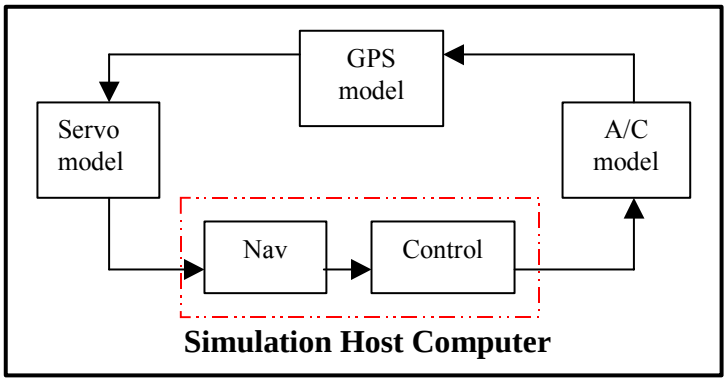

Figure 5 - Software-In-The-Loop configuration

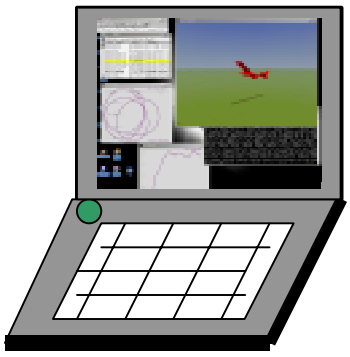

Figure 6 – Software in the loop simulation is often operated on a laptop, to facilitate use in the field during flight testing

Hardware in the Loop

For the HITL simulation, the onboard computer is plugged into a desktop computer via two serial ports. This configuration is illustrated in figure 7. Here, the hardware under test is the onboard computer, along with all software that executes on it. The GPS model and servo interface models provide the proper interfaces to the onboard computer, so the onboard computer configuration is identical to that used in a flight test, shown in figure 8.

The HITL simulation is used to test all guidance, navigation, and control algorithms software and as much of the hardware as practical, in real-time. This configuration is used each time any modification is made concerning either the onboard software or the onboard computer hardware, and prior to any new flight test.

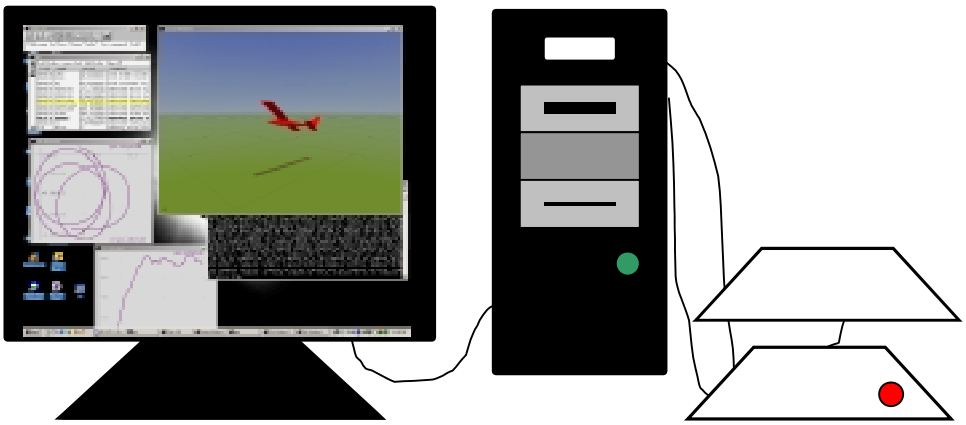

Figure 7 – Hardware in the Loop Simulation

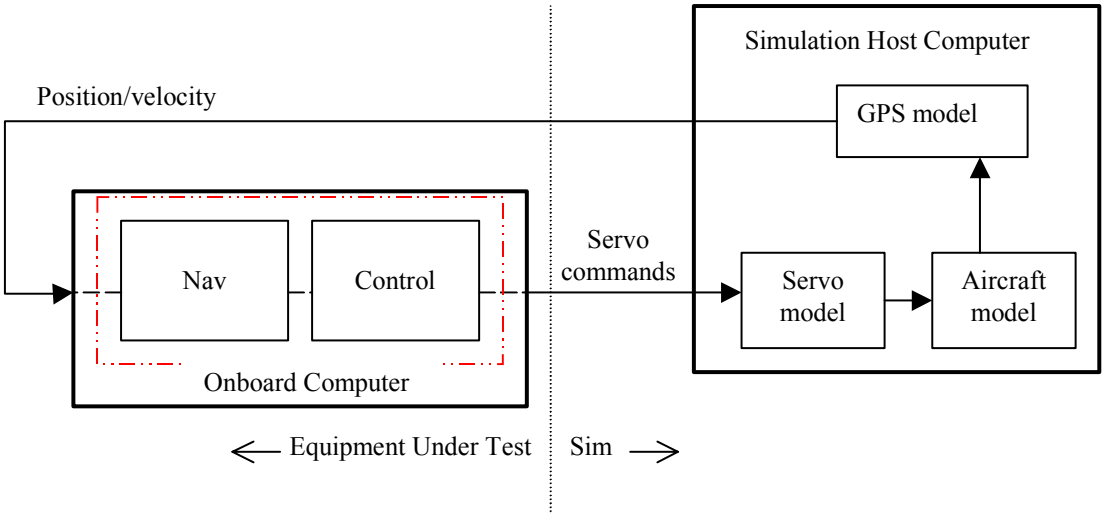

Figure 8 - Hardware-In-The-Loop Simulation

Results

Representative results from the HITL simulation are given in figures 9 and 10. The flight includes an automatic takeoff and then flight to a target point, which the vehicle then circles. The target altitude is 200 feet, with apparent low-frequency error largely due to GPS errors, shown in Figure 9. The first two turns about the point "circularize" the turns about a point, shown in figure 10.

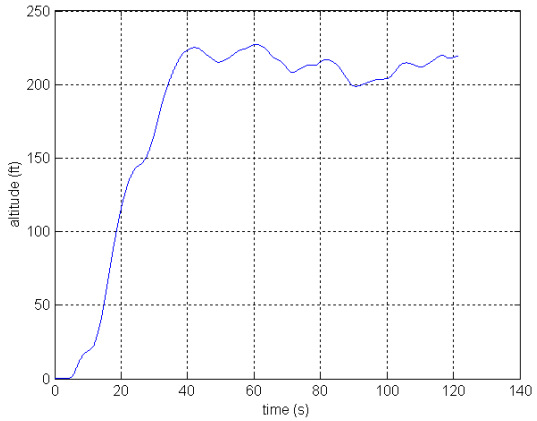

Figure 9 – Altitude time history from hardware in the loop simulation

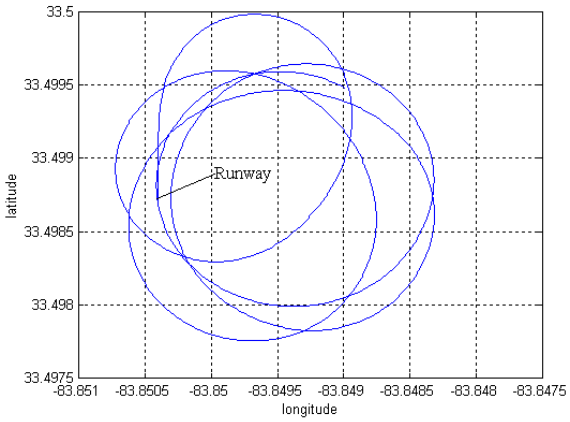

Figure 10 – Position plot from hardware in the loop simulation, the aircraft takes off to the North on the left side of the plot and begins circling about a prescribed point

Conclusions

The flight simulation tools developed are low-cost, utilizing a single personal computer platform. hardware-in-the-loop capabilities are utilized where there is a high benefit for minimal effort. Software-in-the-loop (cross compiling embedded software on test hardware) allows software testing without access to flight hardware. It was also important to have simulation tools that are portable, and usable at the flight testing site. Even though the marginal cost of additional flight testing of a low cost UAV is small, simulation tools remain essential to keeping the amount of flight testing and risk manageable.

References

[1]Kornfeld, R., *The Impact of GPS Velocity Based Flight Control on Flight Instrumentation Architecture*, Massachusetts Institute of Technology Ph.D. Thesis, 1999.

[2]Etkin, B. and Reid, L., *Dynamics of Flight, Stability and Control,* Wiley, 1996.

[3]Nelson, R., *Flight Stability and Automatic Control*, McGraw Hill, 1989.

[4]*Zodiac Serial Data Interface Specification*, Rockwell Semiconductor Systems, Revision 11, 1996.