MINIMUM COMPLEXITY UNINHABITED AIR VEHICLE GUIDANCE AND FLIGHT CONTROL SYSTEM

Eric N. Johnson, Sébastien G. Fontaine, and
School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia

Aaron D. Kahn
Naval Research Laboratory, Arlington, Maryland

Abstract

The availability of commercial single antenna GPS units at low cost and discontinuation of selective availability of the system has cased an increased interest in flying a stable fixed-wing aircraft using GPS alone. Utilizing such an inexpensive sensor, along with a relatively simple processor, a flight control and guidance system could be developed that would be so inexpensive as to be practically disposable even for some commercial applications. A flight control and guidance system that can operate on single antenna GPS measurements is also a candidate as an ultimate backup mode for any uninhabited air vehicle or piloted airplane given failures of sensors. In this paper, necessary hardware and software developments will be described, as well as particular solutions explored in a flight test program.

Introduction

Today, computers with significant computational capabilities continually become less expensive. The economics of consumer electronics has also made inexpensive GPS units available. A remarkable notion is to develop a guidance and flight control architecture that utilizes a single antenna GPS as a sole sensor[1], resulting in a very low cost system, still capable of rudimentary waypoint navigation, altitude control, and approach courses. However, when developing a flight control system for an Uninhabited Air Vehicle (UAV) using GPS alone, several issues arise. These inexpensive commercial units typically have update rates of 1 Hz and significant latency. Also, position and velocity is less information than that conventionally used for flight control, where air data and gyroscopic instruments are commonplace. The challenge is develop algorithms for navigation and flight control which maximize the robustness, tracking bandwidth, and accuracy given this limited sensor capability. An architecture is proposed, which utilizes observers for bank angle and wind components to reconstruct this “missing” sensor information. The airplane is assumed to be statically stable, and with a short-period mode that is fast compared to the response of the autopilot. By these assumptions, angle of attack is effectively set directly by elevator position at a given airspeed. This architecture has been implemented on a small fixed-wing UAV shown figure 1.

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)

This model aircraft used for flight test is a Thunder Tiger Trainer 60 Almost Ready to Fly (ARF) training remote control model airplane with a 0.6 cubic inch engine. During the flight tests the onboard computer is installed under the wing on the right side, the power box under the wing on the left side, and the GPS receiver above the wing as shown figure 1. The total weight of the computer is 1.75 lbs and the total weight of the plane with the flight computer installed is approximately 8 lbs.

Using a commercially available GPS receiver and PC-based processor. This UAV is able to perform guidance and control functions by relying on the reconstruction (observer) of missing sensor signals and on a relatively stable open-loop aircraft. Results from hardware-in-the-loop simulation and flight tests are presented.

A diagram of the complete test system is illustrated in figure 2. Here, the computer, power, radio receiver, and GPS receiver boxes are shown, and the included components shown below the line. The following sections describe the algorithms running on the onboard computer, and the hardware design of the remaining components. This is followed by a description of the testing methods used and results.

Algorithms Implemented

Guidance and flight control algorithms use information given by the GPS to enable the vehicle to fly a prescribed flight plan utilizing aileron, elevator, and throttle control. The rudder is simply set to a fixed position. Note that the system does not have a telemetry system, so flight data is recorded in onboard memory.

The system enables the aircraft to fly to selected waypoints at a selected altitude and speed. Knowing the position of the aircraft and the position of the target waypoint, a track angle to reach this waypoint is calculated. It is then necessary to achieve this desired track angle to reach the waypoint laterally and desired flight path angle to reach the desired altitude.

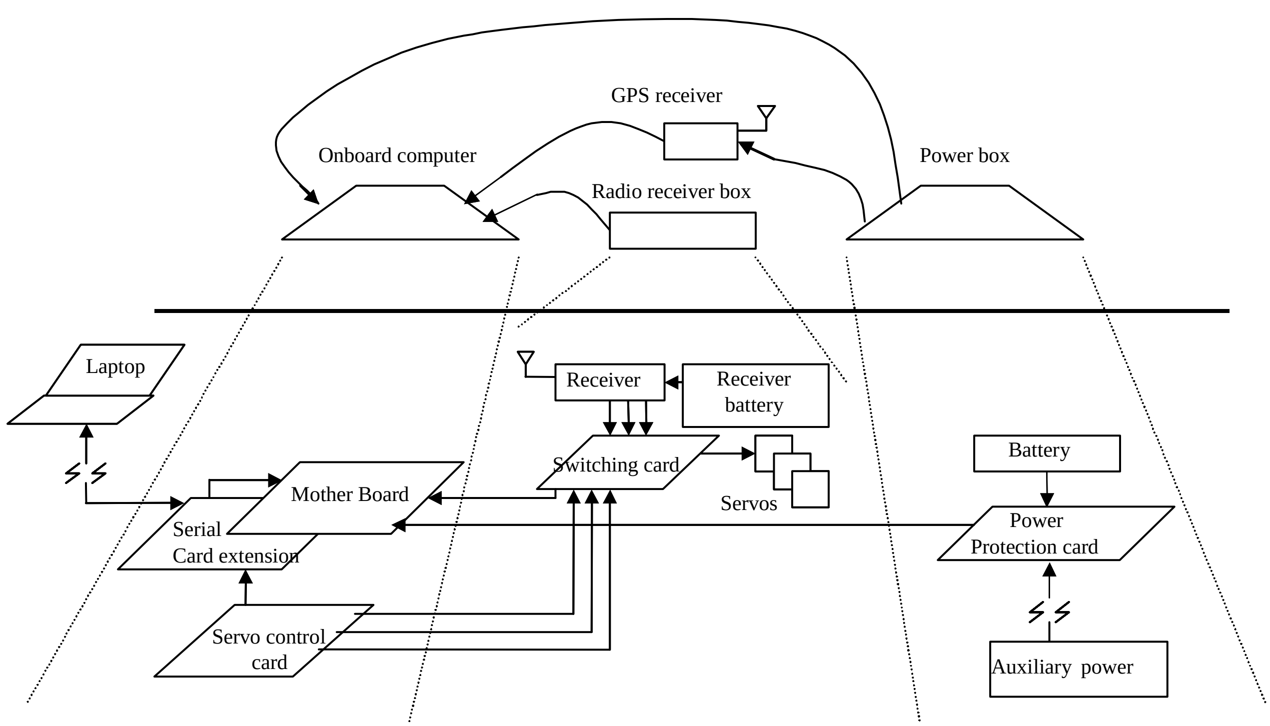

Figure 2. Low cost avionics system architecture

Lateral

Considering a rolling aircraft in a North-East-Down (NED) reference system and assuming:

- Sideslip angle (β) near zero
- Flight path angle (γ) small
- Zero wind (concurrent work is exploring the removal of this assumption through online estimation of wind)

The raw output of a typical off-the-shelf GPS unit has position and velocity ($\vec{v}$) information that can be put in the NED frame. Acceleration ($\vec{a}$) in this frame can be estimated by back differencing velocity information or with an observer. Then,

$$a_{lat} = \frac{\vec{a}_{NE} \times \vec{v}_{NE}}{V_{GS}}$$

is the acceleration of the vehicle in the horizontal (North and East) plane that is perpendicular to the aircraft velocity, where $\vec{v}_{NE}$ and $\vec{a}_{NE}$ are the horizontal components of the velocity and acceleration respectively, $V_{GS} = |\vec{v}_{NE}|$ is the aircraft ground speed, and has only a single component in the down direction. An estimate of the bank angle is:

$$\tilde{\phi} = \arctan\left(\frac{a_{lat}}{V_{GS}(g - a_D)}\right)$$

where a_D is the local down component of vehicle acceleration. This signal is treated as a measurement of bank angle, and is used as an input to a linear fixed-gain observer for bank angle (ϕ).

The commanded bank angle is derived from ground track angle error (difference between direct track to next waypoint and GPS output, $\Delta\psi$, turning in the shortest direction). The desired horizontal acceleration to turn is found by

$$a_{latDes} = \frac{V_{GS}^2}{R}\frac{\Delta\psi}{2\pi}$$

where R is a selected turn radius for the vehicle. A consequence of this choice is that the aircraft will circle the final waypoint with radius R after arrival. This acceleration is then converted to a commanded bank angle by a relation similar to that used above

$$\phi_{des} = \arctan\left(\frac{a_{latDes}}{V_{GS}(g - a_D)}\right)$$

A desired rate of change for bank is found by

$$\dot{\phi}_{des} = \frac{\phi_{des} - \phi}{\tau_\phi}$$

where τ_ϕ is a selected time constant associated with bank angle tracking. Aileron deflection is found by using a single degree of freedom rolling motion analysis[2,3]

$$I_x \dot{p} = c_{l_{\delta_a}} \delta_a + c_{l_p} \frac{pb}{2V}$$

where conventional nomenclature for non-dimensional stability and control derivatives is used[2,3]. For small flight path angles, one can take roll rate p to approximate the time derivative of roll angle $\dot{\phi} \approx p$, and use

$$\delta_a = K_a\left(\dot{\phi}_{des} - p_I\right)\frac{C_{l_p} b}{C_{l_{\delta_a}} 2V}$$

where K_a is a selected gain to relate roll rate error to aileron servo command. The use of non-dimensional stability and control derivatives provides a correction for changes in airspeed, as an alternative to gain lookup tables. The integral term, p_I, is updated by

$$\dot{p}_I = \frac{\phi_{des} - \phi}{\tau_\phi^2} K_{Ia}$$

where K_{Ia} is a non-dimensional gain associated with the integral term. This gives zero steady state error to bank angle tracking.

Longitudinal

In steady state flight, one can take

$$T = D + W\sin(\gamma)$$

where T is the current thrust, D the drag, and W the weight. Considering only small flight path angles (γ)

$$\gamma = \frac{T - D}{W},$$

or for a propeller-driven aircraft

$$\gamma = \frac{P - DV}{WV},$$

where P is the power of the engine, V the aircraft speed, and DV is the power needed to fly the aircraft at a constant altitude. This relationship is used to adjust the throttle setting to account for desired changes in altitude, or a non-zero flight path angle, and to adjust commanded flight path angle to remain within the limits of the engine. This is done by assuming power required is a function of the velocity and power used at a nominal flight condition (say, $\delta_{t_{nom}}$ is 50% throttle with an associated steady-level speed of V_{nom}), or

$$DV = \frac{V}{V_{nom}} \delta_{t_{nom}} P_{\max}$$

Note that a more complex relation for the way power is related to throttle and speed could be utilized. With the above choice, one obtains the following estimate for the flight path angle that would result from any combination of speed and throttle setting:

$$\tilde{\gamma} = \frac{P_{\max}}{WV}\left(\delta_t - \delta_{t_I} - \delta_{t_{nom}} \frac{V}{V_{nom}}\right).$$

where δ_{t_I} is an integral term on throttle, discussed further below. It is included here to ensure that the relation is exact in the steady state. So, flight path angle is increased when throttle is increased, or speed decreased.

To achieve a commanded altitude, a commanded flight path angle is found by

$$\gamma_{des} = \frac{h_{des} - h}{\tau_h V}$$

where h_{des} is the commanded altitude, h is the corresponding GPS output, and τ_h is a selected time constant associated with altitude capture. To further assure that the throttle will never be pushed beyond totally closed or totally open, the following upper and lower limits on commanded flight path angle are imposed

$$\gamma_{\min} = \frac{P_{\max}}{WV}\left(5\% - \delta_{t_I} - \delta_{t_{nom}} \frac{V}{V_{nom}}\right)$$

$$\gamma_{\max} = \frac{P_{\max}}{WV}\left(95\% - \delta_{t_I} - \delta_{t_{nom}} \frac{V}{V_{nom}}\right)$$

To achieve the desired speed V_{des}, the following acceleration signal is determined

$$\dot{V}_{des} = \frac{V_{des} - V}{\tau_V}$$

where τ_V is a selected time constant associated with speed capture. Throttle position to be used is then found by

$$\delta_t = \frac{mV}{P_{\max}}\left(\dot{V}_{cmd} + g\gamma_{des}\right) + \frac{V}{V_{nom}}\delta_{t_{nom}} + \delta_{t_I}$$

where the first term on the right hand side allows throttle to control airspeed with a correction for a non-zero flight path angle. The second term adjusts for changes in commanded airspeed. The final term is an integral term, updated by

$$\dot{\delta}_{t_I} = \frac{mV}{P_{\max}}\left(V_{cmd} - V\right)\frac{K_{It}}{\tau_V^2}$$

where K_{It} is a non-dimensional integral gain constant. The throttle command is position and rate limited to prevent harm to the engine.

To determine elevator angle, a desired angle of attack is first found. To get this signal a commanded flight path rate is found by

$$\dot{\gamma}_{des} = \frac{\gamma_{des} - \gamma}{\tau_\gamma}$$

where τ_γ is a selected time constant. The angle of attack that will give this flight path angle rate is approximately

$$\alpha_{des} = \frac{\dot{\gamma}_{des} V m}{\cos(\phi) c_{L_\alpha} \bar{q} S} + \alpha_I$$

where $\overline{q}$ is an estimate of dynamic pressure based on GPS speed, S is the wing area, and α_I is an integral term (i.e., estimate of angle of attack for level flight).

One can use the approximation for pitching moment coefficient in trimmed flight

$$c_m = 0 = c_{m_o} + c_{m_\alpha}\alpha + c_{m_{\delta_e}}\delta_e + c_{m_q}\frac{q\overline{c}}{2V}$$

Solving for elevator, for the desired angle of attack

$$\delta_e = \frac{-1}{c_{m_{\delta_e}}}\left(c_{m_\alpha}\alpha_{des} + c_{m_q}\frac{q\overline{c}}{2V}\right)$$

where c_{m_o} has been accommodated by the integral term. Assuming that pitch rate is coming from a steady-level turn[2,3] this becomes

$$\delta_e = \frac{-1}{c_{m_{\delta_e}}}\left(c_{m_\alpha}\alpha_{des} + c_{m_q}\frac{\overline{c}g\sin^2(\phi)}{2V^2}\right)$$

The angle of attack integral term is updated by

$$\dot{\alpha}_I = \frac{mV(\gamma_{des} - \gamma)}{\cos(\phi)c_{L_\alpha}\overline{q}S\tau_\gamma^2}K_{Ie}$$

where K_{Ie} is an integral term gain. The resulting controller achieves zero steady state altitude tracking error.

Hardware Design

As illustrated in figure 2, the flight hardware consists of a GPS antenna/receiver and a safety-pilot radio receiver system plugged into a computer. The computer consists of a main processor and a serial card extension. A single battery powers the GPS (requiring 5 VDC at 150 mA,), the computer, the servo interface, and the serial card requiring (5 VDC at 1.2 A), a DTXC 2100 flat NI-CAD battery delivering 7.2 VDC at 1500 mA is used to power the system.

The choices made concerning the flight computer were evaluated using two principle criteria: low cost and weight. After an initial search, two possible solutions for most components were found. Two candidates for the processor, where a MC68332 and an Intel 80386SX. Two candidates for the GPS receiver, where the CMC Superstar and the Delorme's Earthmate.

The final choice for the GPS was the Delorme Earthmate[4] and the chosen configuration for the flight computer was:

- AR-9612B mother board
- 80386Sx Intel processor 4Mb RAM
- 720 KB of flash memory
- A diamond systems emerald serial card with 4 serial ports RS-232
- VXworks 5.3 as operating system

The flight computer is managing a pontac SV200 servo interface, chosen because of its high flexibility concerning the system used to switch between the manual and the automatic flight.

The power distribution system, figure 3, is a simple regulation circuit, which manages the current providing, coming from a battery or any admissible auxiliary power, even at the same time. It has two power outputs one for the GPS, the other for the computer.

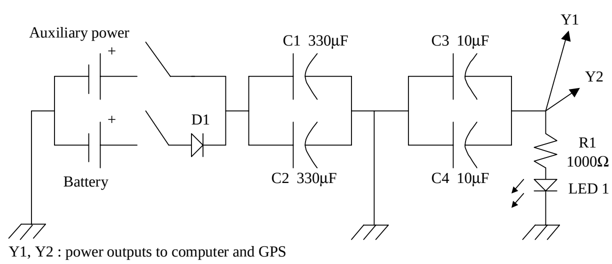

Figure 3. Power distribution system

The switch box is the piece of custom electronics, which allows us to switch the autopilot on and off. As shown on figure 4 the inputs from the computer and the output going to the SV200 are all optically isolated. There are two optical isolators in each box Q1 and Q2, and hence a total of four isolators, three are used for the actuator commands from the SV200 to the servos and one is used to send the auto/manual switch position from the switch box to the SV200. The isolators allow switching voltage without direct electrical connections between circuits. Therefore it acts as interference protection because both receiver and switch are electrically isolated from the computer.

The way the transistors are wired to these optical isolators will invert the signal. To correct this a Schmitt trigger inverter is used, Q3 in the figure. Using this trigger instead of a more common one ensures a transistor-to-transistor logic (TTL) signal even if any slope appears on the input signal.

An element is needed to physically make the input data to the servos switch between the radio receiver (safety pilot manual) and the computer (automatic). This is the role of Q4 in figure 4. Q4 is a 2-1 multiplexor. This acts like four single pole double throw switches, each one representing one channel of the multiplexor. Only three out the four channels are used here. There are three inputs into each channel and only one output. For each channel two of the three inputs are for the signal to switch and the third input is to signal to switch. All of the computer signals are sent to I1a, I1b, and I1c, and the signals from the receiver are sent to I0a, I0b, and I0c. The commands come through Za, Zb, and Zc, which are sent to the servos. When the auto/manual switch is thrown on the safety pilot transmitter, the three controls (ailerons, elevator, throttle) are switched at the same time. The signal that makes the switch, called S, is tied to Q5 in the figure. Q5 is a PIC needed to command to the multiplexor to make the switch. On the S line, The PIC turns either a line high, or low, depending on the position of a switch on the safety pilot radio control transmitter. The output is tied to a LED and to an optical-isolator as well as to the line S. The LED shows when the switching process is on and the line going through the optical-isolator is sent to one of the SV200 A/D converter, which tells the computer if the autopilot is on. This is subsequently used by the computer to determine if integral terms should be updated.

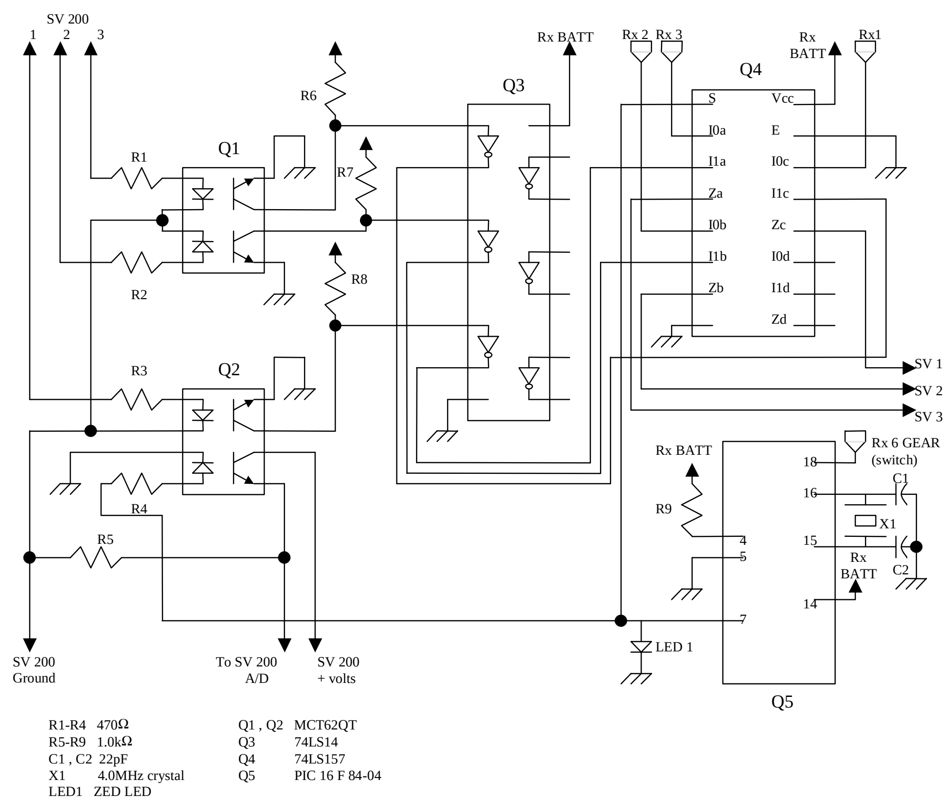

Figure 4. Automatic and manual (safety pilot) switching device

Test Configurations

Three test configurations have been utilized for this system: software in the loop, hardware in the loop[5] and actual flight tests.

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

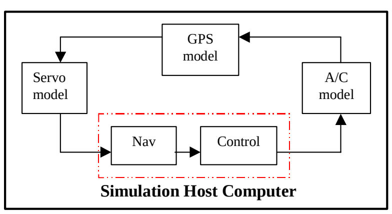

Figure 5. Software-In-The-Loop configuration

Hardware in the loop

For the HITL simulation, the onboard computer is plugged into a desktop computer via two serial ports. 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[4] to the onboard computer, so the onboard computer configuration is identical to that used in a flight test, shown in figure 6.

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.

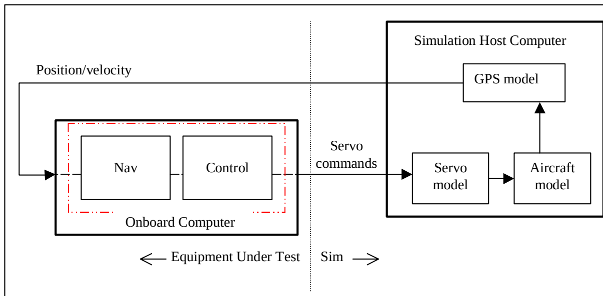

Figure 6. Hardware-In-The-Loop Simulation

Flight tests configuration

When performing flight tests, the final low-cost avionics system is configured, with the addition of a manual safety pilot override. Using this system the performance of this low cost flight control system can be evaluated. The configuration is illustrated in figure 7, which is equivalent to figure 2.

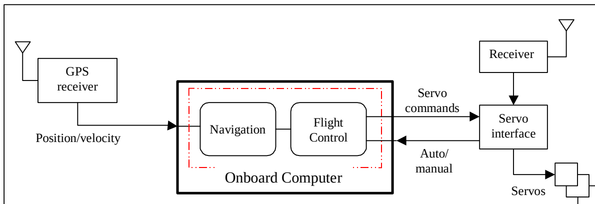

Figure 7. Flight Test Configuration

Test Results

Representative results from the HITL simulation are given in figures 8 and 9. 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 9. For actual flight tests, the takeoff and landing are flown manually by the safety pilot.

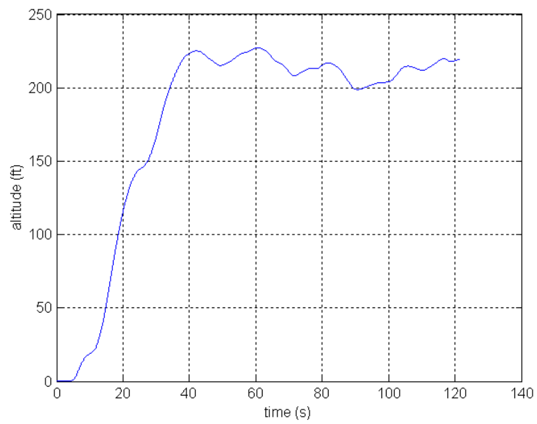

Figure 8. Altitude time history from hardware in the loop simulation

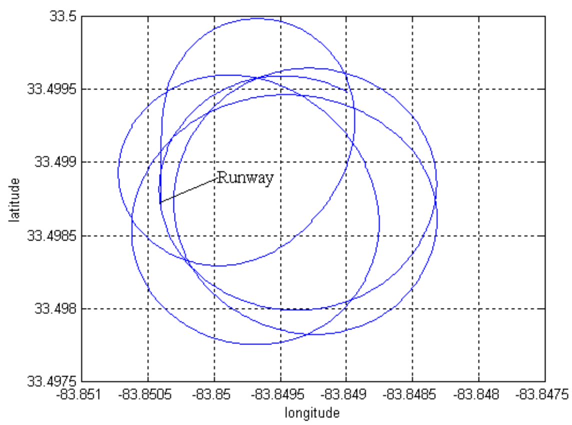

Figure 9. Position plot from hardware in the loop simulation, the aircraft takes off to the North on the left side of the plot and begins circling about a prescribed point

The performance of the GPS has been evaluated in flight test for a manual flight by the safety pilot. Figure 10 is a plot of the latitude and longitude during this flight.

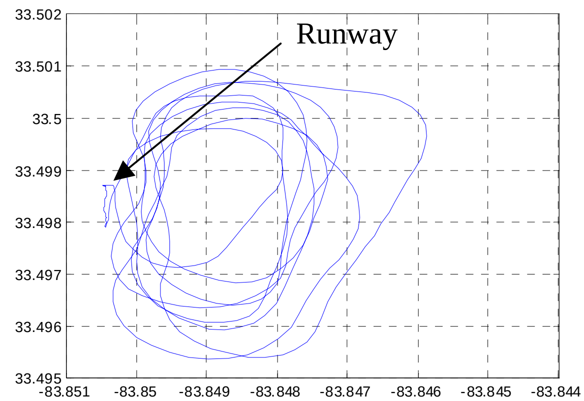

Figure 10. Recoreded flight test GPS data from a manual flight, performing a race track pattern

CONCLUSIONS

An avionics system based solely on the use of GPS for the navigation and flight control has been designed, developed, and is being tested. The simulation tools are also 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. Future work includes further flight-testing of the system, incorporation of wind estimation and improved autopilot functionality. Utilizing such an inexpensive architecture, along with a relatively simple processor, a flight control and guidance systems can be developed that would be extremely inexpensive. A flight control and guidance system that can operate on single antenna GPS measurements is also a candidate as an ultimate backup mode for any uninhabited air vehicle or piloted airplane given failures of other sensors.

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.

[5]Johnson, E. and Fontaine, S., "Use of flight simulation to complement flight testing of low-cost UAVs," *AIAA Modeling and Simulation Technologies Conference*, 2001.