Software- and Hardware-in-the-Loop Verification of Flight Dynamics Model and Flight Control Simulation of a Fixed-Wing Unmanned Aerial Vehicle*

Calvin Coopmans [†]   Michal Podhradský   Nathan V. Hoffer

Abstract

Unmanned aerial system (UAS) use is ever-increasing. In this paper, it is shown that even with low-cost hardware and open-source software, simple numerical testing practices (software- and hardware-in-the-loop) can prove the accuracy and usefulness of an aeronautical flight model, as well as provide valuable pre-flight testing of many situations typically only encountered in flight: high winds, hardware failure, etc. Software and hardware simulation results are compared with actual flight testing results to show that these modeling and testing techniques are accurate and provide a useful testing platform for a small unmanned aerial vehicle. Source code used in simulation is open and provided to the community.

1. INTRODUCTION

Currently, unmanned aerial systems (UAS) are experiencing exponential growth for civilian applications. One of the key issues facing UAS is how to integrate them into civil airspace, and as with any airborne system, safety is the highest priority. For unmanned aerial vehicles (UAVs), safety means obstacle avoidance, communication between manned and unmanned aircraft, and robust and fault tolerant systems [1],[2], & [3].

Beginning in 2006, AggieAir has been researching various civil applications of unmanned aerial vehicles, focusing on water-related scientific data collection and analysis. Through a service center at the Utah Water Research Laboratory (UWRL) at Utah State University, AggieAir utilizes both vertical takeoff and landing (VTOL) and Fixed-wing platforms to research the collection important data about crop health, stream habitat,

Figure 1. AggieAir Fixedwing Platform – Minion, during landing maneuver

invasive species, etc. To this end, data collection is the mission of AggieAir, and many system-level factors affect the quality of the data missions such as safety, reliability, stability, etc. Overall, AggieAir is a small, low-cost, autonomous, multispectral remote sensing platform [4]; a current AggieAir fixed-wing UAS platform is shown in Fig. 1, and a current hexarotor VTOL platform is shown in Fig. 2. For flight control, the open-source Paparazzi autopilot system [5] is used for all platforms, with low-cost electronics and navigation sensors, while maintaining excellent flight characteristics and reliability [6].

1.1. Motivation

Balancing safety and system complexity while retaining cost-effectiveness is a challenging task. Testing is crucial for the success of any system, and properly planned tests are key to showing that a design (such as a control system) conforms to its specifications. Since there are risks associated with any use of airspace, the ability to run tests in a controlled environment such as a virtualized one is invaluable to enabling exploration of new UA system elements such as control and navigation algorithms, etc. Since data collection for scientific in-

---

Figure 2. AggieAir Multirotor Platform – Ark, ready for takeoff

formation is the mission of AggieAir, improvements in these system design elements can provide gains in data quality which can be crucial in enabling new applications and better scientific outcomes.

One common technique for demonstrating the functionality of flight control systems is software- and hardware-in-the-loop (SITL and HITL). This is a testing scheme during which a flight control system (autopilot) is applied to a numerical model of the target aircraft, while the control algorithms or autopilot hardware respectively run as though in a real flight. Other systems such as failsafes and payloads can also be tested in-the-loop, and edge cases such as hardware/software failure can be applied to a designed system in a safe and controlled environment, leading to better overall robustness and reliability, i.e. safety.

1.2. Contribution

This work's main contribution is to show how model-based development of a small UAS can proceed even with low-cost tools such as an open-source UAS autopilot (i.e. Paparazzi), and that when executed properly, testing and verification techniques such as hardware-in-the-loop can be used to show robustness of the system in varying circumstances and thus derive the benefits of advanced testing. Even small, low-cost UAS such as the AggieAir Minion platform used in this work can benefit from aerospace testing rubrics, which contribute to greater reliability and higher levels of safety for all entities in the airspace.

It is the author's intent to make the SITL and HITL source code freely available for any user of the Paparazzi community, in order to enable more develop-

ers to write more fully tested, safer autopilot code. The work associated with this publication will be made available as part of the Paparazzi project.

1.3. The Minion Airframe

The AggieAir Minion platform is a fixedwing T-tail configuration small UAS. The aircraft has been optimized for endurance and for takeoff and landing without a prepared surface. The Minion platform has been designed around a multispectral sensor scientific payload but can be adapted to custom payloads. The configuration of the AggieAir Minion platform is given in Table 1.

Table 1. Summary of the AggieAir Minion UAV configuration

Parameter	Value	Units
As flown configuration	4.1	kg
Maximum weight	8.16	kg
Maximum payload weight	1.81	kg
Wing span (low load)	2.29	m
Wing span (high load)	2.74	m
Fuselage length	1.08	m
Range	54	km
Flight time	1	hr
Cruse speed	15	m/s
Typical land survey area	1200	acres

Flight characteristics and wind affect the quality of collected data. Trajectory tracking and disturbance rejection are the main characteristics needed from the flight control system. PID controllers, are currently used for the Minion platform and are simple, well proven, and have adequate trajectory tracking and disturbance rejection for AggieAir's data collection missions.

1.4. Paparazzi Unmanned Aerial System

Paparazzi [5] is an open-source autopilot. Its main advantage over similar systems (such as Pixhawk or ArduoPilot) is high modularity and configurability–it is easy to customize for needs of the user. AggieAir has been using Paparazzi for over seven years with good success.

Paparazzi, similarly to other open source autopilots, uses PID control loops for attitude and altitude stabilization as well as navigation control (path following). Although more advance control approaches have been successfully used on UAVs, PID control is still the most common in the world of small UAVs. The reason is that PID controllers do not require a system model to

be successfully deployed (and a missing system model is often the case), and can be easily tuned *manually* i.e. trial and error method when starting with stable gains. PID controllers provides sufficiently good flight performance and have been proven over many flight hours. Because outside academia there is little incentive for "fixing what isn't broken," UAV developers can rarely justify costs (i.e. crashing) required to develop a more sophisticated control system without high-quality testing tools.

By *sufficiently good flight performance* we mean accurate trajectory following with good disturbance rejection. It is difficult to provide exact numbers since the requirements depend on the data mission. For the purpose of this paper, however, we require a stable flight with oscillations dampened within 10 seconds.

1.5. Existing Modeling and Simulation Work

Hardware-in-the-loop simulations for manned aircraft have been a standard aerospace procedure for years. Now with more computing power and modeling software available, HITL is being used for UAVs. Guowei et al. [7] use HITL to verify control system performance for a helicopter UAV and give results of the simulation vs. flight data. Dongwon et al. [8] use a HITL simulation to verify avionics and the control system. Their HITL uses actuator models identified through system identification and aerodynamic coefficients estimated with Digitial DATCOM. Johnson et al. [9] use HITL simulation to verify software and hardware changes prior to flight. The simulation is also used in the field before control gains are changed verify control system performance.

In this paper, Sec. 2 gives background on aeronautical models and model choice for small UAS testing. Section 3 shows two kinds of flight mission simulation (software- and hardware-in-the-loop), and 4 shows how real flight data of a Minion aircraft match well with the expected simulation results.

2. MINION FLIGHT DYNAMIC MODELS

Hardware-in-the-loop simulation requires an accurate model of the UAV flight dynamics, which can be modeled using equations of motion. The Minion platform is modeled using the general linearized 6-Degree of Freedom equations of motion found in [10] shown generally in Eq 1:

$$\dot{x} = ax + bu. \qquad (1)$$

Within a and b are forces and moments equations, an example of which is given in Eq. 2

$$\frac{1}{2}\rho V^2 S_w C_Z, \qquad (2)$$

where C_Z is the force coefficient in the z body direction of the aircraft. The set of force and moment coefficients within Eq. 1 must be determined to accurately model the flight dynamics of the Minion platform. There are 4 main methods for determining the force and moment coefficients. These are numerical methods such as lifting-line theory [11], wind tunnel testing [12], system identification [13], and CFD [14]. The reasons for using one or the other depends on time, budget, equipment, required precision, and so on.

For this particular case, the lifting-line method is used to predict the force and moment coefficients. Lifting-line theory, developed independently by Frederick W. Lanchester and by Ludwig Prandtl, predicts the lift distribution across lifting surfaces. By varying the angle of attach, side slip angle, velocity, and control surface deflection, the force and moment coefficients can be estimated, then these coefficients are used to define the equations of motion. The estimated force and moment coefficients for the Minion aircraft are given in Table 2 and are used in JSBSim [15] to simulate the dynamics of the aircraft.

3. UNMANNED SIMULATION OF MODEL AND CONTROLS

Simulation of flight for the Minion platform was performed in two ways: software-in-the-loop (SITL), and hardware-in-the-loop (HITL). Both simulations were carried out at a dedicated test station shown in Fig. 3. Both simulations were carried out using the same flight plan, as were the actual flight tests detailed later in this paper.
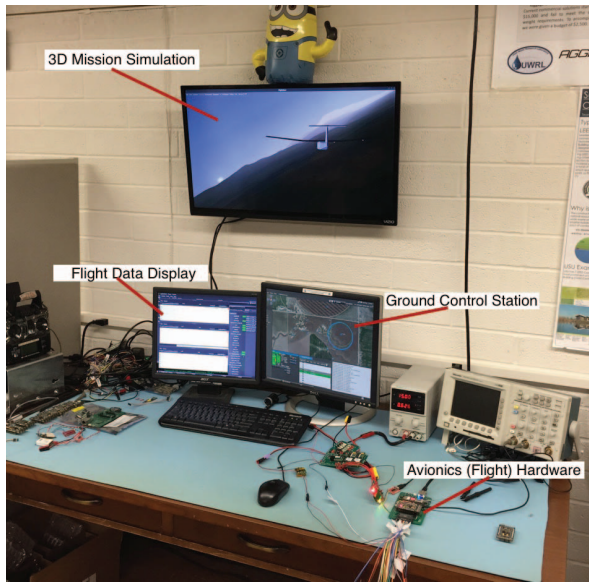
3.1. Simulation Software Toolset

JSBSim is an open source flight dynamic model (FDM), used in many UAV-related projects (Aerocross Echo Hawk, DuPont Aerospace Company, MITRE Air Traffic Studies, U.S. Department of Transportation, NASA and, indeed, the popular flight simulator Flight-Gear) and is considered to be one of the best open source solutions for FDM. JSBSim has been also used previously in Paparazzi for modeling smaller and lighter (foam core flying wings) UAVs. JSBSim uses a linearized flight model, based on lift, drag and other coefficients of the aircraft. JSBSim has been used for modeling the F-16, Cessna, and even the Space Shuttle, as

Table 2. Summary of the force and moment coefficients

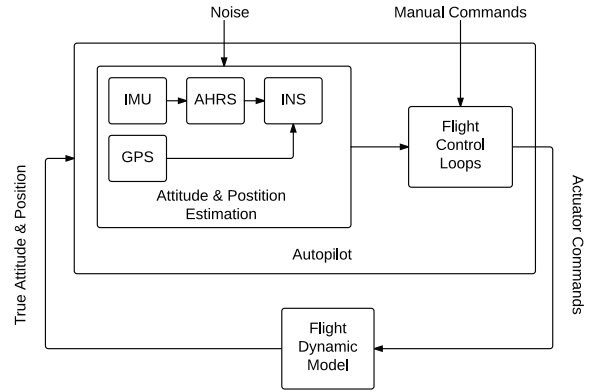|		
$C_X = -0.026$	$C_Y = 0$	$C_Z = -0.403$
	$C_m = -0.014$	
$C_{X,u} = 0$	$C_{Z,u} = 0$	$C_{m,u} = 0$
$C_{X,\alpha} = -0.139$	$C_{Z,\alpha} = -5.619$	$C_{m,\alpha} = -1.550$
$C_{Y,\beta} = -1.575$	$C_{\ell,\beta} = -0.236$	$C_{n,\beta} = 0.468$
$C_{Y,p} = -0.3219$	$C_{\ell,p} = -0.6490$	$C_{n,p} = 0.0230$
$C_{X,q} = -0.0934$	$C_{Z,q} = -4.1141$	$C_{m,q} = -10.6883$
$C_{Y,r} = 1.0089$	$C_{\ell,r} = 0.2331$	$C_{n,r} = -0.3128$
$C_{Y,\delta a} = 0.0341$	$C_{\ell,\delta a} = -0.4258$	$C_{n,\delta a} = -0.0238$
$C_{X,\delta e} = -0.0026$	$C_{Z,\delta e} = -0.4676$	$C_{m,\delta e} = -1.2071$
$C_{Y,\delta r} = 0.3152$	$C_{\ell,\delta r} = 0.0573$	$C_{n,\delta r} = -0.1429$

Figure 3. SITL/HITL test station in "flight."

3.2. Software-in-the-loop simulation

Figure 4. Software-in-the-loop (SITL) autopilot testing block diagram

well as previously mentioned UAVs: is well suited for the small UAS simulations.

The XML configuration file is used to define areas and coefficients of the aircraft, based on the lifting-line theory numbers in Table 2. Then the definitions for the propeller and motor are added to complete the simulation model. The simulation itself is called from a C++ program together with other flight data, such as initial position, altitude, and environmental information (wind direction).

It is the author's intent to make the SITL and HITL source code freely available for any user of the Paparazzi community, in order to enable more developers to write more fully tested, safer autopilot code. The work associated with this publication will be made available as part of the Paparazzi project [5].

Software-In-The-Loop (SITL) (Fig. 4) is simulation containing the autopilot control code and attitude & position estimation code, but compiled and run on a laptop or a desktop computer. The simulation is run against FDM of the airplane, so the response of the airplane model, the autopilot control loops and the position/attitude estimation algorithms (if applicable) can be observed.

The advantage of SITL is that it is easy to deploy and test, because no additional hardware is needed. This means that simulation is very self-contained. After developing a Minion FDM, SITL is used to tune the control gains for autonomous flight. This is considered successful since the simulation flew the model autonomously through various flight maneuvers like those expected to be encountered in the field.

3.3. Hardware-in-the-loop simulation
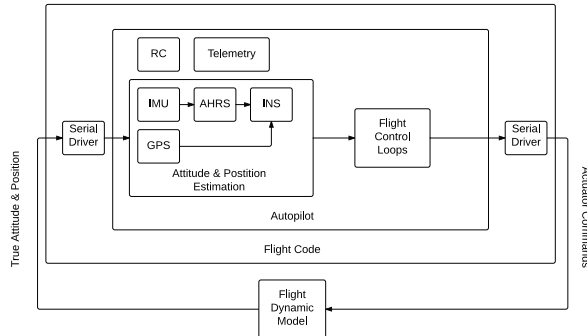
Figure 5. Hardware-in-the-loop (HITL) autopilot testing block diagram

Hardware-In-The-Loop (Fig. 5) runs actual flight code on the autopilot hardware (flight code and flight hardware), simulating the sensor inputs (in this case serial packets) and FDM response. Hardware-in-the-loop is the simulation closest to real flight, because both the hardware and the code are identical to the set being used in real flight–the autopilot is really flying with artificial sensor data.

Indeed the challenge is to have a precise FDM, because the closer the FDM is to the real aircraft, the closer HITL simulation is to real flight. It is possible to even simulate changing battery voltage (based on the power used by motor), airspeed etc. using either replacements for digital sensors or having DAC converters providing the desired value, guided by the simulator.

In the ideal case the flight performance in SITL and HITL would be identical, but that is often not the case for several reasons. First, the timing in HITL is very important–the autopilot has to receive the right data at the right moment, and the FDM has to receive control inputs from the autopilot at the right time. In SITL there is more leeway, since the timing is only pseudo real-time (SITL can "catch up" or "slow down" the simulation as needed). Second, the code in HITL includes drivers, hardware abstraction layer (HAL) and other control-unrelated code which might cause slightly different responses (rounding errors on 32bit vs 64bit platforms, for example).

4. VERIFICATION OF FLIGHT DY-NAMICS MODEL AND CON-TROLLER USING FLIGHT DATA

This section outlines the process of verifying the flight dynamics model of the Minion simulation platform, the PID flight controller, and gains used in fully autonomous flight. Following this process ensures software and hardware are nearly flight tested and control gains are reasonably well tuned. The actual flight test was conducted under the authorization of the FAA under COA 2014-WSA-88.

4.1. Model Verification

To provide functional verification of the HITL, the following was performed:

1. Developed FDM of Minion airframe using:

   - lifting–line theory: to determine force and moment coefficients
   - 3D modeling: to determine moments of inertia and center of gravity
   - physical measurements: to determine weight of the plane

2. Iteratively tested the developed FDM in manual simulated flight (SITL with RC transmitter)

   - based on human pilot input, we adjusted uncertain model parameters until the perceived response was consistent with pilot's flight experience

3. Tuned autonomous control loop gains in SITL for two modes:

   - AUTO1 - is a control mode with augmented stability. Pilot commands the desired roll and pitch angles, and the autopilot corrects the airplane to follow.
   - AUTO2 - is a fully autonomous mode, the pilot has no control authority

4. Perform a fully autonomous SITL flight (take-off, flight, landing)

   - This verifies that the autopilot control loops operate correctly

5. Perform a fully autonomous HITL flight (take-off, flight, landing)

   - This verifies that the flight code performs as desired and that the flight code and hardware is ready for flight

- Note that the flight plan is the same in SITL, HITL and real flight

6. Perform an autonomous flight

   - with manual take-off and landing (for safety purposes, since the flight area has rough terrain)
   - the flight maneuvers are, however, identical to SITL/HITL simulations, as described below.

4.2. Flight Maneuvers
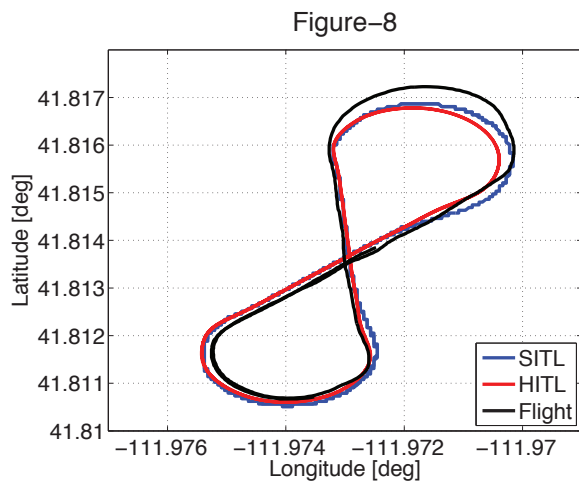
Figure 6. Figure-8 test results: SITL, HITL, and Minion flight data.

Figure-8 Maneuver. Consists of flying in a "figure-8"-like pattern around a given waypoint. This maneuver is important, because most of the time the airplane is in maximal roll (maximal allowed roll for autonomous flight, typically around 30 degrees), banking into the turn which makes *figure-8* a good benchmark of altitude following during turns. The plane should hold its altitude while maintaining the desired turn radius (in this case 120 meters AGL). A *figure-8* also has short linear transitions between turns, during which the airplane is required to stabilize (i.e. no longitudinal oscillations) and follow the desired altitude. Figure 6 shows test results with SITL, HITL, and Minion flight data showing good autonomous tracking of the *figure-8* test pattern, despite 6 m/s wind conditions.

In addition to path following, it is important that the HITL and flight data match in time. For the HITL to truly represent reality, the time it takes to complete a maneuver or parts of a maneuver should be the same.
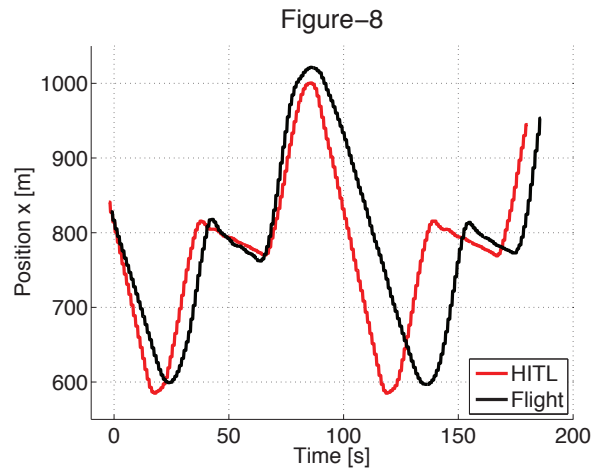
Figure 7. Figure-8 test results: x position vs time of HITL and Minion flight data.
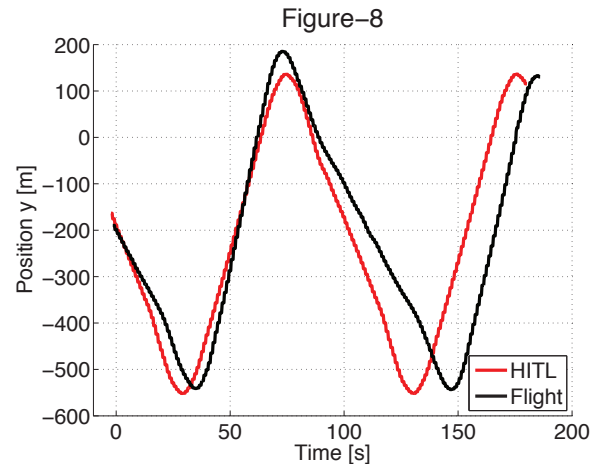
Figure 8. Figure-8 test results: y position vs time of HITL and Minion flight data.

It is possible for the path following to be correct while the time it takes to complete the maneuver to be very different between HITL and flight. The x and y position vs. time for the *figure-8* is shown in Figs. 7 and 8. The manuever timing is similar up until 120 seconds where the aircraft completes the turn more quickly than HITL simulation. This difference in timing can be contributed to wind disturbances. The HITL and flight data are within 10% of each other.

Line 1-2 Maneuver. Is a simple line between two waypoints with the same altitude. This maneuver is most common during regular missions, when the plane scans
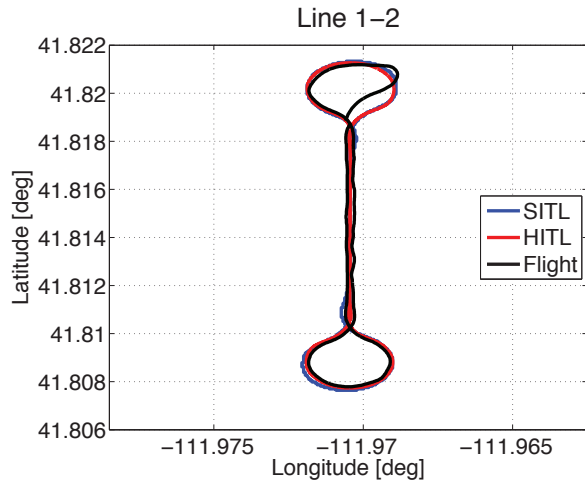
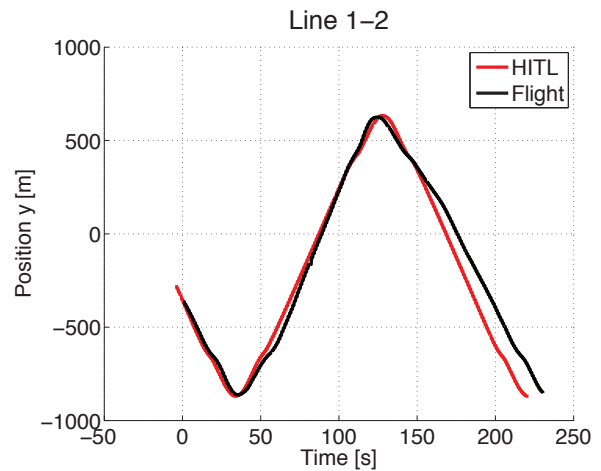**Figure 9. Line 1-2 test results: SITL, HITL, and Minion flight data.**

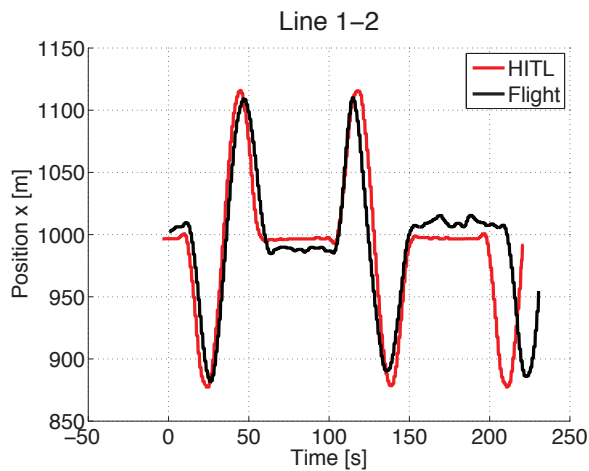**Figure 10. Line 1-2 test results: X-position vs time of HITL and Minion flight data.**

Figure 11. Line 1-2 test results: Y-position vs time of HITL and Minion flight data.

5. CONCLUSIONS

The main contribution of this paper is in demonstrating the usability of hardware-in-the-loop testing for low-cost, open-source, small civilian UASs. We have shown an overview of a sufficiently accurate Flight Dynamic Model of a small UAV using lifting-line theory. We then tested the autopilot control loops using the developed FDM in software-in-the-loop simulation. After proving that the autopilot can maintain a stable autonomous flight, we proceeded to test the performance of complete avionics in HITL simulation assuming no wind, etc. The autopilot was again able to maintain a stable autonomous flight in HITL mode. Finally, we validated our simulation with a real flight, using identical control system and identical flight plan. All three "flights" were remarkably similar (within 10% error, despite a windy flight day), which leads to conclusion that the FDM and both SITL and HITL are sufficiently accurate and can be used for rapid development and tuning of control loops, as well as testing of new autopilot extensions (i.e. additional navigation routines or intelligence).

The AggieAir group and the authors believe in Open Source software, and are proudly making the code for SITL and HITL simulation publicly available, so UAV developers and researchers can verify their control systems before flight, thus increasing safety and reliability of UAV operation.

Future work will include incorporating of weather data in the simulation, and a better battery model. Then we will be able to accurately simulate not only flight performance, but also influence of wind and mission

the area of interest and takes pictures. Thus it is important to have a very stable flight during *line 1-2* maneuver. We are also looking for good trajectory following in presence of side wind (crabbing). Typically if the plane performs well in *figure-8* it will perform well in *Line 1-2*. Figure 9 shows the Minion SITL, HITL, and autonomous flight *line 1-2* performance: excellent tracking of the requested flight plan despite 6m/s wind conditions.

Figures 10 and 11 show the time-position comparison of HITL and flight data. For *Line 1-2* the maneuver timing is more consistent than the *figure-8*.

duration. Our vision is that in near future the UAS operators will be able to plan next days mission, having an accurate FDM model of the UAS, weather forecast, and data coverage requirement. Then operators will simulate the mission, and optionally change it and simulate again until a satisfactory flight plan is developed. The subsequent UAV flight will be predicted by the simulation, and will perform as well as predicted. Thus speeding up the planning and pre-flight check process, while maintaining high safety and reliability of airborne operations.

ACKNOWLEDGMENT

References

[1] Federal Aviation Administration, "Integration of civil unmanned aircraft systems (UAS) in the national airspace system (NAS) roadmap," FAA, Tech. Rep., 2013.

[2] K. Dalamagkidis, K. P. Valavanis, and L. Piegl, "On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the national airspace system," *Progress in Aerospace Sciences*, vol. 44, no. 7-8, pp. 503–519, Oct. 2008.

[3] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, "Current status and future perspectives for unmanned aircraft system operations in the US," *Journal of Intelligent and Robotic Systems*, vol. 52, no. 2, pp. 313–329, Feb. 2008.

[4] A. M. Jensen, Y. Chen, M. McKee, T. Hardy, and S. L. Barfuss, "Aggieair - a low-cost autonomous multispectral remote sensing platform: new developments and applications," in *Proceedings of the 2009 IEEE International Geoscience and Remote Sensing Symposium*, vol. 4, July 2009, pp. IV–995–IV–998.

[5] Paparazzi Forum, "Open-source Paparazzi UAV project." [Online]. Available: http://paparazzi.enac.fr/

[6] C. Coopmans, B. Stark, and C. M. Coffin, "A payload verification and management framework for small UAV-based personal remote sensing systems," in *Proceedings of the 2012 Int. Symposium on Resilient Control Systems (ISRCS2012)*. IEEE, 2012, pp. 184–189.

[7] G. Cai, B. M. Chen, T. H. Lee, and M. Dong, "Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters," *Mechatronics*, vol. 19, no. 7, pp. 1057–1066, 2009.

[8] J. Dongwon and T. Panagiotis, "Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle," in *AIAA 2007 Conference and Exhibit*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 2007, pp. 1–13.

[9] E. Johnson and S. Fontaine, "Use of flight simulation to complement flight testing of low-cost UAVs," in *Proc. of the 2001 AIAA Modeling and Simulation Technologies Conference*, 2001, pp. 1–6.

[10] W. F. Phillips, *Mechanics of Flight*, 2nd ed. Wiley, 2009.

[11] W. F. Phillips and D. O. Snyder, "Modern Adaptation of Prandtl's Classic Lifting-Line Theory," *Journal of Aircraft*, vol. 37, no. 4, pp. 662–670, 2000.

[12] W. Wisnoe, R. Effendy, M. Nasir, W. Kuntjoro, and A. Mohd, "Wind Tunnel Experiments and CFD Analysis of Blended Wing Body (BWB) Unmanned Aerial Vehicle (UAV)," in *Proc. of the 13th International Conference on Aerospace Sciences & Aviation Technology (ASAT-13)*, 2009.

[13] M. B. Tischler and R. K. Remple, *Aircraft and rotorcraft system identification: engineering methods with flight test examples*. Virginia: Amer Inst of Aeronautics & Astronautics, 2006.

[14] W. F. Phillips, D. F. Hunsaker, and R. J. Niewoehner, "Estimating the Subsonic Aerodynamic Center and Moment Components for Swept Wings," *Journal of Aircraft*, vol. 45, no. 3, pp. 1033–1043, 2008.

[15] "JSBSIM Project Homepage." [Online]. Available: http://jsbsim.sourceforge.net/