Modeling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle

Dongwon Jung* and Panagiotis Tsiotras†

Georgia Institute of Technology, Atlanta, GA, 30332-0150

Modeling and experimental identification results for a small unmanned aerial vehicle (UAV) are presented. The numerical values of the aerodynamic derivatives are computed via the Digital DATCOM software using the geometric parameters of the airplane. Flight test data are utilized to identify the stability and control derivatives of the UAV. The aerodynamic angles are estimated and used in conjunction with inertial measurements in a batch parameter identification algorithm. A hardware-in-the-loop (HIL) simulation environment is developed to support and validate the UAV autopilot hardware and software development. The HIL simulation incorporates a high-fidelity dynamic model that includes the sensor and actuator models, from the identified parameters from experiments. A user-friendly graphical interface that incorporates external stick commands and 3-D visualization of the vehicle's motion completes the simulation environment. The hardware-in-the-loop setup is an indispensable tool for rapid certification of both the avionics hardware and the control software, while performing simulated flight tests with minimal cost and effort.

I. Introduction

Unmanned aerial vehicles (UAVs) have received considerable attention in both civilian and military applications. For complex missions, UAVs need to integrate both hardware and software in a seamless manner. For small UAVs the limited payload constraint impedes the use of high performance avionics systems, equipped with complete inertial and air data sensors. A successful certification of a new autopilot can only be enabled by extensive flight testing to debug both the hardware and the software before deploying the UAV in the field. Despite the necessity of real flight tests, simulation-based testing also plays a very important role. It can save time and effort prior to conducting actual flight tests. In particular, hardware-in-the-loop (HIL) simulation testing should be adopted whenever possible to validate together both the hardware and the software under realistic conditions.[1,2]

Building a simulation model starts with setting up the equations of motion for a 6-DOF dynamic model. Although the equations of motion for a fixed-wing aircraft are well known and can be found in many textbooks,[3,4] difficulties arise from handling the aerodynamic forces and moments. Traditionally, the dimensionless coefficients for the aerodynamic forces and moments are approximated by a linear sum of contributing parameters utilizing the stability and control derivatives in a specified flight condition. Precise knowledge of such derivatives is essential towards the development of a high fidelity simulation. Wind tunnel test can be utilized to obtain these derivatives experimentally, but this is a laborious task demanding not only enormous experimental data sets but also great effort in order to analyze them to identify the derivatives for a certain flight condition. On the other hand, the geometric parameters of the aircraft can be used to provide a good estimate of these derivatives.[5,6] This approach has been successfully applied to the real aircraft.[7]

Another approach for identifying the stability and control derivatives is to use actual flight data. Identification of the stability and control derivatives directly from flight test data has been a subject of interest in the aerospace community for a ling time.[8] Recently, Refs. 9,10 presented a recursive parameter identification algorithm for these derivatives so that a reconfigurable control algorithm adapts to actuator failure according to parameter variations. Frequency domain identification methods have been discussed in Refs. 11,12. Reference 13 summarizes some identification methods that are directly applicable to aircraft.

[*] Graduate student, School of Aerospace, dongwon_jung@ae.gatech.edu, AIAA Student Member.

[†] Professor, tsiotras@gatech.edu, AIAA Associate Fellow.

This article describes a complete simulation environment which has been developed at Georgia Tech for testing and validating UAV control systems. Here the aerodynamic stability and control derivatives are obtained either from the geometric data of the airplane or from flight test data. A batch identification method is presented to estimate the stability and control derivatives from real flight data, incorporating estimated angle of attack and sideslip angle. The details for building a comprehensive HIL simulation are presented for the high fidelity nonlinear modeling, sensor and actuator subsystem modeling, 3-D visualization, remote pilot command interface, and the seamless interconnection between the hardware and the software components. The HIL simulation environment allows not only testing the hardware and software but also conducting simulated flight tests with minimal cost and effort.

II. Equations of Motion of a Fixed-Wing Aircraft

The standard 6-DOF equations of motion for a conventional aircraft are used for modeling and simulation of a small UAV. Flat Earth approximation[3] provides a reasonable modeling assumption when the vehicle operates over a small area. The body-axes equations are as follows:

Force equations:

$$\begin{aligned} \dot{U} &= rV - qW - g\sin\theta + (X_A + X_T)/m, & \text{(1a)}\\ \dot{V} &= -rU + pW + g\sin\phi\cos\theta + (Y_A + Y_T)/m, & \text{(1b)}\\ \dot{W} &= qU - pV + g\cos\phi\cos\theta + (Z_A + Z_T)/m, & \text{(1c)} \end{aligned}$$

Moment equations:

$$\begin{aligned} J_x\dot{p} - J_{xz}(\dot{r} + pq) + (J_z - J_y)qr &= \bar{L}, & \text{(2a)}\\ J_y\dot{q} + (J_x - J_z)pr + J_{xz}(p^2 - r^2) &= M, & \text{(2b)}\\ J_z\dot{r} - J_{xz}(\dot{p} - qr) + (J_y - J_x)pq &= N, & \text{(2c)} \end{aligned}$$

Kinematic equations:

$$\begin{aligned} \dot{\phi} &= p + \tan\theta(q\sin\phi + r\cos\phi), & \text{(3a)}\\ \dot{\theta} &= q\cos\phi - r\sin\phi, & \text{(3b)}\\ \dot{\psi} &= (q\sin\phi + r\cos\phi)/\cos\theta, & \text{(3c)} \end{aligned}$$

Navigation equations:

$$\begin{aligned} \dot{p}^N &= U\mathrm{c}\theta\mathrm{c}\psi + V(-\mathrm{c}\phi\mathrm{s}\psi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi) + W(\mathrm{s}\phi\mathrm{s}\psi + \mathrm{c}\phi\mathrm{s}\theta\mathrm{c}\psi), & \text{(4a)}\\ \dot{p}^E &= U\mathrm{c}\theta\mathrm{s}\psi + V(\mathrm{c}\phi\mathrm{c}\psi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi) + W(-\mathrm{s}\phi\mathrm{c}\psi + \mathrm{c}\phi\mathrm{s}\theta\mathrm{s}\psi), & \text{(4b)}\\ \dot{p}^D &= -U\mathrm{s}\theta + V\mathrm{s}\phi\mathrm{c}\theta + W\mathrm{c}\phi\mathrm{c}\theta, & \text{(4c)} \end{aligned}$$

where the definition of each variable is found in Ref. 3.

The aerodynamic forces and moments are obtained from the dimensionless aerodynamic coefficients at a given flight condition as follows,

$$\begin{aligned} X_A = \bar{q}SC_X, \quad Y_A = \bar{q}SC_Y, \quad Z_A = \bar{q}SC_Z, && \text{(5a)}\\ \bar{L} = \bar{q}SbC_l, \quad M = \bar{q}S\bar{c}C_m, \quad N = \bar{q}SbC_n. && \text{(5b)} \end{aligned}$$

The analysis of the aerodynamic behavior of the aircraft, however, is better understood in the stability axes, or wind axes, system. In particular, the aerodynamic forces are easily handled in the wind axes system, which is given in terms of the angle of attack α and the sideslip angle β. Hence, the force equations in wind-axes are introduced as follows,[3]

$$\begin{aligned} m\dot{V}_T &= T\cos(\alpha + \alpha_T)\cos\beta - D + mg_1, & \text{(6a)}\\ m\dot{\beta}V_T &= -T\cos(\alpha + \alpha_T)\sin\beta - C_w + mg_2 - mV_T r_s, & \text{(6b)}\\ m\dot{\alpha}V_T\cos\beta &= -T\sin(\alpha + \alpha_T) - L + mg_3 + mV_T(q\cos\beta - p_s\sin\beta), & \text{(6c)} \end{aligned}$$

where, V_T is the total air speed, p_s and r_s are the pitch and yaw rates projected on the wind axes, m is the mass of the vehicle, and g_1 and g_2 are the projections of the gravitational acceleration on the wind axes system. The aerodynamic forces are lift (L), drag (D), and cross wind force (C_w) while T is the thrust force exerted on the aircraft.

II.A. Aerodynamic coefficients for forces and moments

The aerodynamic forces and moments have a complex dependence on a number of variables resulting in complicated nonlinear correlation between each variable. Building up the aerodynamic forces and moments as a sum of contributing components provides a mathematically convenient way of representing the aerodynamic forces and moments for a specified flight condition. By the same token, dimensionless aerodynamic coefficients, which are associated with the stability and control derivatives in terms of independent parameters, have been widely used to represent the aerodynamic characteristic of an aircraft:

$$\begin{aligned}
C_D &= C_{D0} + \frac{(C_L - C_{L0})^2}{\pi e AR} + |C_{D_{\delta e}} \delta_e| + |C_{D_{\delta_a}} \delta_a| + |C_{D_{\delta r}} \delta_r|, && \text{(7a)}\\
C_Y &= C_{y_\beta}\beta + C_{y_{\delta_a}}\delta_a + C_{y_{\delta_r}}\delta_r + \frac{b}{2V_T}(C_{y_p} p_s + C_{y_r} r_s), && \text{(7b)}\\
C_L &= C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\delta_e}}\delta_e + \frac{\bar{c}}{2V_T}(C_{L_{\dot\alpha}}\dot\alpha + C_{L_q} q), && \text{(7c)}\\
C_m &= C_{m0} + C_{m_\alpha}\alpha + C_{m_{\delta_e}}\delta_e + \frac{\bar{c}}{2V_T}(C_{m_{\dot\alpha}}\dot\alpha + C_{m_q} q), && \text{(7d)}\\
C_l &= C_{l_\beta}\beta + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r + \frac{b}{2V_T}(C_{l_p} p_s + C_{l_r} r_s), && \text{(7e)}\\
C_n &= C_{n_\beta}\beta + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r + \frac{b}{2V_T}(C_{n_p} p_s + C_{n_r} r_s), && \text{(7f)}
\end{aligned}$$

where, e is the Oswald coefficient and AR is the main wing aspect ratio.

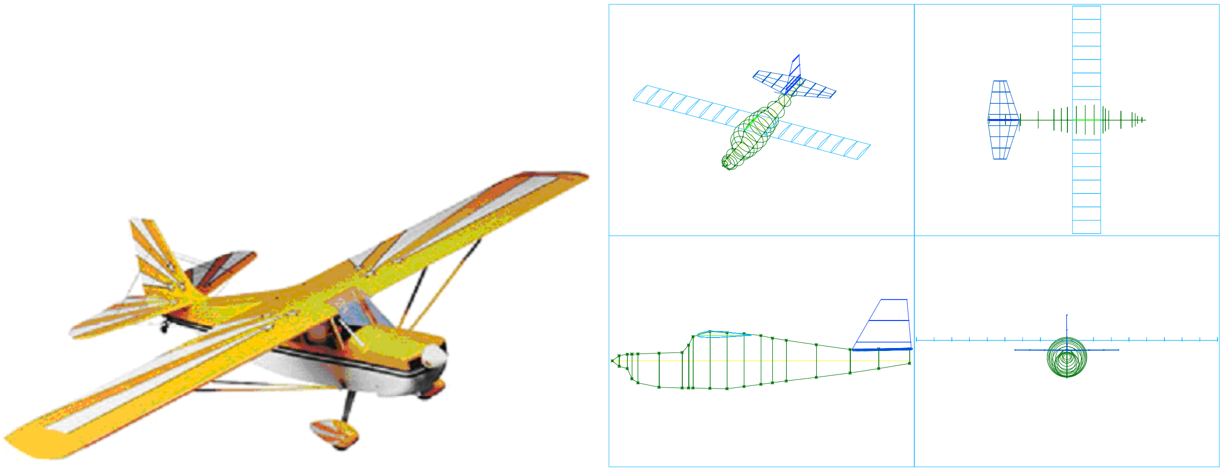
II.B. Numerical modeling of aerodynamic derivatives

The stability and control derivatives shown in Eq. (7) determine the aerodynamic characteristic of the aircraft, and can be estimated or identified through wind tunnel tests or via flight tests. Despite the fact that the wind tunnel test gives accurate results, a complete solution for all derivatives is not always feasible from wind tunnel test alone. Furthermore, it is beneficial to obtain these derivatives from empirical data, if possible. The United States Air Force (USAF) has developed the stability and control data compendium (DATCOM),[5] a large collection of information containing classical aerodynamic analysis and experimental data. The digital DATCOM[7] is the digital version of DATCOM, a computer program which was originally written in Fortran and ported afterwards to ANSI C. It incorporates the classical aerodynamic equations with empirical corrections from experimental data for various aircraft to compute the aerodynamic stability and control derivatives of the aircraft. The DDATCOM is useful in predicting the stability and control characteristics for preliminary aircraft designs or developing a high fidelity flight simulator.[14] The DDATCOM takes an input file containing the aircraft configuration and geometric parameters, the flight condition, the mass properties, and so on. The input configuration file of the UAV used in this research is shown in Table 1.

The geometric configuration of the UAV airframe was accurately measured and entered in the DDATCOM input file. This included the total weight, the shape of the wing/tail airfoil section, a cylindrical modeling of the fuselage section, the center of gravity location, the placement of wing/tail, etc. Figure 1(a) shows the UAV airframe, a commercial Decathlon R/C model, and Fig. 1(b) shows the detailed CAD model of the airframe. The total weight of the UAV was measured using a precise weight scale, and the center of gravity location was carefully determined in the body axes with respect to the base point at the center of the propeller. The mass moments of inertia about each principal axis were identified from a torsional pendulum experimental setup.[15] With the geometric data supplied, DDATCOM computes the static stability derivatives, the dynamic stability derivatives, and control derivatives for both the aileron and the elevator. The control derivatives associated with the rudder and the stability derivatives associated with the yawing moment are not computed from the DDATCOM. These derivatives can be estimated using the method proposed in Ref. 16. The stability and control derivatives calculated from the DDATCOM are summarized in Table 2.

Table 1. Digital DATCOM input configuration file for the 1/5 scale Decathlon.

Properties	value	units
Mass (m)	5.6132	[kg]
Aerodynamic reference area (S)	0.6558	[m^2]
Longitudinal reference length ($\bar{c}$)	0.3215	[m]
Lateral reference length (b)	2.04	[m]
Free stream airspeed (V_T)	20	[m/sec]
Flight path angle (γ)	0	[deg]
Altitude (h)	300	[m]



(a)

(b)

Figure 1. Geometric modeling of the 1/5 scale Decathlon used as an input to the DDATCOM program.

II.C. Thrust modeling

The UAV is powered by a two-stroke internal combustion engine with a fixed pitch propeller. As a complete modeling of the engine with the propeller is beyond our scope, a simplified modeling of the thrust generated is given as follows. Assume that the engine is overpowered providing enough torque to the propeller. Hence the thrust delivered by the propeller is solely dependent on the aerodynamics of the propeller so the thrust force can be calculated based on blade element theory.[17,18] For a fixed geometry propeller, the thrust force varies according to the rotating speed of the propeller. This static thrust is computed from the computer program JavaProp,[19] provided the geometry of the propeller. In addition to the thrust calculation, an experiment was conducted to identify the actual thrust at different rotating speeds. A spring scale was used to measure the static thrust along a set of throttle commands that results in different rotating speeds. A correction to the calculated static thrust was done by comparing with the actual thrust measurement by taking into account the reduced mass flow rate over the downstream region due to the fuselage hindrance.

The dynamic thrust, on the other hand, is quite complicated to be estimated directly from propeller geometry. In general, the thrust force during flight depends not only on the rotational speed of the propeller but also on the forward flight speed (incoming wind speed). The forward flight speed reduces the generated thrust force by the propeller. This aspect will continue until a certain combination of RPM and forward flight speed at which no thrust is generated and changes its sign for an opposite direction (wind mill effect). In order to take into account this aspect, the advance ratio (J) should be introduced to incorporate the effect of forward flight speed as follows

$$J = \frac{\pi V_T}{\Omega R_p}, \tag{8}$$

where R_p is propeller radius in [m] and Ω the rotational speed in [rad/sec]. From the analysis using JavaProp

Table 2. Static and dynamic stability derivatives and control derivatives of the 1/5 scale Decathlon.

C_{D_0}	C_{L_0}	C_{L_α}	$C_{L_{\dot{\alpha}}}$	C_{L_q}	C_{y_β}	C_{y_p}	C_{y_r}
0.028	0.062	5.195	1.22	4.589	-0.2083	0.0057	0.0645*
C_{m_0}	C_{m_α}	$C_{m_{\dot{\alpha}}}$	C_{m_q}	C_{l_β}	C_{l_p}	C_{l_r}	C_{n_β}
0.0598	-0.9317	-2.897	-5.263	-0.0377	-0.4625	0.0288	0.0116
C_{n_p}	C_{n_r}	$C_{D_{\delta_e}}$	$C_{D_{\delta_a}}$	$C_{D_{\delta_r}}$	$C_{L_{\delta_e}}$	$C_{y_{\delta_a}}$	$C_{y_{\delta_r}}$
-0.0076	-0.0276	0.0418	$0.0^\dagger$	$0.0^\dagger$	0.2167	$0.0^\dagger$	0.1096*
$C_{m_{\delta_e}}$	$C_{l_{\delta_a}}$	$C_{l_{\delta_r}}$	$C_{n_{\delta_r}}$	$C_{n_{\delta_a}}$			
-0.8551	-0.2559	$0.0085^\dagger$	0.0035	$-0.0216^\dagger$			

† : approximately zero assumed, * : estimation from the Reference 16.

and with the help of the correction from the actual thrust, the thrust coefficient (c_T) was obtained in terms of the advance ratio (J) as shown in Fig. 2. This dimensionless coefficient is related to the actual thrust by

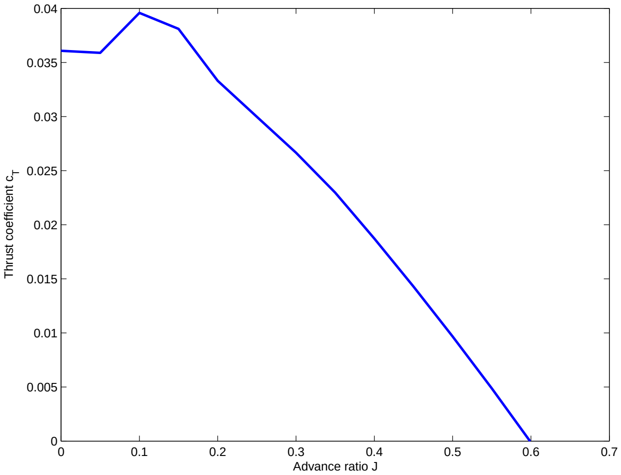


Figure 2. Thrust coefficient c_T v/s advance ratio for the selected propeller.

the following relation,

$$T = \frac{4}{\pi^2}\rho\Omega^2 R_p^4 c_T, \tag{9}$$

where, ρ is the air density. Hence, the actual thrust can be estimated from Eq. (9) using an approximation of c_T at a given advance ratio value that is computed from the flight speed and the rotational speed of propeller at each instant.

III. Estimation of Aerodynamic Angles

The angle of attack and the sideslip angle are significant states describing the aerodynamics of the airplane. Specifically, these angles are required for identification of the aerodynamic force and moment derivatives. For exact measurements, it is necessary to install appropriate air data sensors on-board the airplane. However, it is sometimes difficult to install such apparatuses on a small UAV. Another approach, used here, is to determine the aerodynamic angles using inertial sensor measurements. In this section, the aerodynamic angles are estimated from raw inertial acceleration measurements using an approach similar to the one presented in Refs. 20, 21.

III.A. Filter formulation

Complementary filters have been widely used for combining two independent noisy measurements of a same signal, where each measurement is corrupted by different types of spectral noise.[22] Figure 3 shows a complementary filter that uses two measurements $x_m(t)$ and $\dot{x}_m(t)$ to obtain an estimate $\hat{x}(t)$ of $x(t)$. The time constant τ is selected according to the noise characteristics of each channel such that the estimate $\hat{x}$ is contributed by integration of $\dot{x}_m$ over frequencies $\omega \gg 1/\tau$, whereas for frequencies $\omega \ll 1/\tau$, $\hat{x}$ tracks x_m.

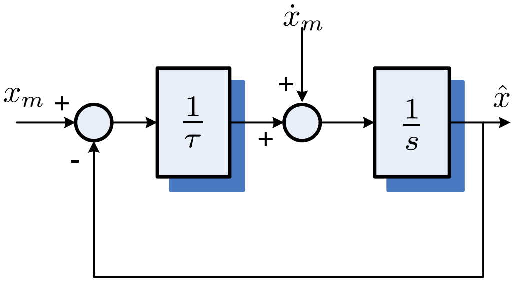


Figure 3. A complementary filter in the feedback form.

In order to employ a complementary filter in the estimation of the aerodynamic angles, one needs to find the aerodynamic angles and their time derivatives computed from the sensor measurements. Using a small angle approximation for α and β during a steady state flight condition, the body z-axis accelerometer output a_m^z measures approximately the thrust force and the aerodynamic force in the wind-axes as follows

$$ma_m^z \cong T\sin(\alpha + \alpha_T) + L. \tag{10}$$

It follows from Eq. (6c) that the time derivative of the angle of attack is obtained by

$$\dot{\alpha}_m = \frac{g - a_m^z}{V_T} + q. \tag{11}$$

The thrust force components in Eq. (10) can be further neglected from the assumption of small angle of attack and the thrust vector aligned to the body x-axis. Knowing that the lift force L is represented by the dimensionless coefficient of Eq (7c), one can obtain the angle of attack from the accelerometer output and the aircraft states using the following relation

$$\alpha_m = \frac{1}{C_{L_\alpha}}\left(\frac{m}{\bar{q}S}a_m^z - \frac{\bar{c}C_{L_{\dot{\alpha}}}}{2V_T^2}(g - a_m^z) - C_{L_0} - C_{L_{\delta_e}}\delta_e - \frac{\bar{c}q}{2V_T}(C_{L_{\dot{\alpha}}} + C_{L_q})\right). \tag{12}$$

It should be noted that a priori knowledge of the stability and control derivatives used in Eq. (12) is important to get a correct estimate of the angle of attack. These parameters can be supplied by the method discussed in Sec. II.B.

For the sideslip angle estimation, the body y-axis accelerometer output a_m^y measures approximately the thrust force and the aerodynamic force as follows

$$ma_m^y \cong T\cos(\alpha + \alpha_T)\sin\beta + C_w, \tag{13}$$

With a small angle approximation of α and β during a steady-state flight condition at small θ, it follows that $g_2 \approx g\sin\phi$ and $r_s \approx r$ in Eq. (6b). Hence the time derivative of the sideslip angle yields

$$\dot{\beta}_m = \frac{1}{V_T}\left(g\sin\phi - a_m^y\right) - r. \tag{14}$$

The thrust force components in Eq. (13) can also be neglected along the same reason discussed above. Assuming that the cross-wind force component C_w in Eq. (6b) is related to the body force component Y_A by $C_w \cong -Y_A$, it follows from Eqs. (7b) that the sideslip angle is obtained from the accelerometer output and the aircraft states using the following relation

$$\beta_m = -\frac{1}{C_{y_\beta}}\left(\frac{m}{\bar{q}S}a_m^y + C_{y_{\delta_r}}\delta_r + \frac{b}{2V_T}(C_{y_p}p + C_{y_r}r)\right). \tag{15}$$

The estimates of the angle of attack and the sideslip angle are then computed from Eqs. (11),(12),(14) and (15) using complementary filters, as illustrated in Fig. 3.

IV. Identification for Aerodynamic Force/Moment Coefficients

In this section, the stability and control derivatives of the UAV are identified from actual flight test data. In contrast to the discussion in Sec. II.A, the aerodynamic coefficients with respect to the body axes system will be dealt with in the following formulation due to the fact that measurements are obtained via body-fixed inertial sensors. As discussed already, these coefficients are assumed to be linear in terms of each contributing component, so a linear parameterization is adopted as the identification model structure.

IV.A. A linear parameterization

The measurements for model identification are provided by the on-board sensor suite. Among these sensors, accelerometers and rate gyros capture the dynamic behavior of the airplane, which yields the aerodynamic force coefficients expressed in the body axes as follows

$$C_X = \frac{m}{\bar{q}S}a_m^x - \frac{T}{\bar{q}S}, \quad C_Y = \frac{m}{\bar{q}S}a_m^y, \quad C_Z = \frac{m}{\bar{q}S}a_m^z, \tag{16}$$

where the thrust force was assumed to be only exerted along the body x-axis and is modeled as a function of the rotating speed of the propeller and the forward flight speed as discussed in Sec. II.C. The aerodynamic moment coefficients are also obtained as follows

$$C_l = \frac{\bar{L}}{\bar{q}Sb}, \quad C_m = \frac{M}{\bar{q}S\bar{c}}, \quad C_n = \frac{N}{\bar{q}Sb}, \tag{17}$$

where $\bar{L}$, M and N are calculated from Eqs. (2a), (2b) and (2c) using the body rate measurements $\boldsymbol{\omega} = [p\;q\;r]^\mathsf{T}$ and the corresponding numerical differentiation $\dot{\boldsymbol{\omega}} = [\dot{p}\;\dot{q}\;\dot{r}]^\mathsf{T}$.

A linear regression model structure was opted for by taking into account the decoupled longitudinal and lateral dynamics of the aircraft. Hence, decoupled longitudinal and lateral derivatives are associated with each aerodynamic coefficient as follows

$$\begin{aligned}
C_X &= C_{x_0} + C_{x_\alpha}\alpha + C_{x_q}\frac{\bar{c}}{2V_T}q + C_{x_{\delta_e}}\delta_e, && \text{(18a)}\\
C_Y &= C_{y_0} + C_{y_\beta}\beta + C_{y_p}\frac{b}{2V_T}p + C_{y_r}\frac{b}{2V_T}r + C_{y_{\delta_a}}\delta_a + C_{y_{\delta_r}}\delta_r, && \text{(18b)}\\
C_Z &= C_{z_0} + C_{z_\alpha}\alpha + C_{z_q}\frac{\bar{c}}{2V_T}q + C_{z_{\delta_e}}\delta_e, && \text{(18c)}\\
C_l &= C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_T}p + C_{l_r}\frac{b}{2V_T}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r, && \text{(18d)}\\
C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{\bar{c}}{2V_T}q + C_{m_{\delta_e}}\delta_e, && \text{(18e)}\\
C_n &= C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_T}p + C_{n_r}\frac{b}{2V_T}r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r. && \text{(18f)}
\end{aligned}$$

The stability and control derivatives are unknown parameters to be identified, so the model structure in Eqs. (18) yields the following linear parametrization for each aerodynamic coefficient,

$$z = \boldsymbol{\theta}^\mathsf{T}\mathbf{x}, \tag{19}$$

where, z is a measurement scalar and can be obtained from Eq. (16) or (17) for each aerodynamic coefficient, $\boldsymbol{\theta}$ is a unknown parameter vector, and $\mathbf{x}$ is a regressor matrix of which entries are comprised of the measurements from sensors along with the right-hand-side of Eqs. (18).

The problem of batch identification is to find $\boldsymbol{\theta}$ that satisfies Eq. (19). A best fit is obtained using a batch process utilizing a set of measurements $z(k)$ and $\mathbf{x}(k)^\mathsf{T}$ as follows

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(1)^\mathsf{T} \\ \mathbf{x}(2)^\mathsf{T} \\ \vdots \\ \mathbf{x}(n)^\mathsf{T} \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} z(1) \\ z(2) \\ \vdots \\ z(n) \end{bmatrix}. \tag{20}$$

The best fit is obtained from the least-square optimization problem,

$$\min_{\boldsymbol{\theta}} \|\mathbf{z} - \mathbf{X}\boldsymbol{\theta}\|^2, \tag{21}$$

whose solution is

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}\mathbf{z}. \tag{22}$$

IV.B. Identification from flight test data

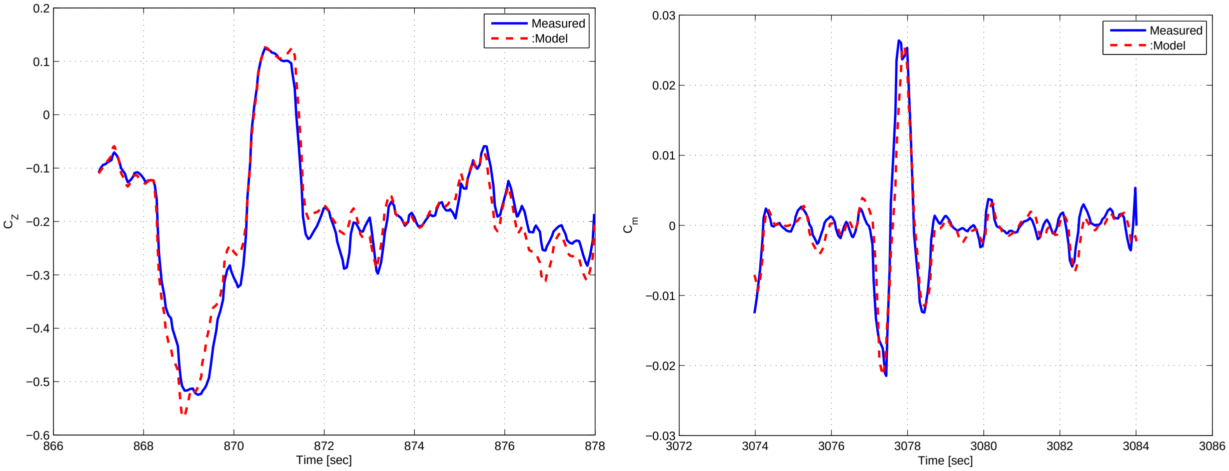
Flight test data were collected during maneuvers initiated from a trim condition of steady and straight level flight by exciting each control surface. Doublet-like open-loop commands to each control surface were issued by a remote pilot for each case, and upon the assumption of decoupled longitudinal and lateral dynamics, the aileron or the rudder doublets were executed for identifying the lateral coefficients (C_Y, C_l, and C_n), whereas the elevator doublet commands were executed for identifying the longitudinal coefficients (C_m, C_X, and C_Z). The sensor measurements were sampled at 20 Hz and were processed a posteriori to remove measurement noises or biases. For estimating the aerodynamic angles and attitude angles for building the regressor matrices, the estimation filters which were discussed in Sec. III and Ref. 23 were utilized for off-line calculation of those variables.

Several data sets were selected for each identification process to assure the correctness of the results. Subsequently, the identified parameters are found to be consistent in each case within a reasonable bound of the estimation error. Table 3 shows the identified parameters for longitudinal and lateral derivatives. Note that the error value given inside the brackets represents the 1-σ standard deviation of the estimate error which provides the confidence interval of the estimate. The longitudinal coefficients result mostly in good agreement as the numerical models were validated against experiment data taken from different maneuvers shown in Fig. 4. The validation of the lateral coefficients are shown in Fig. 5.

Table 3. Identified stability and control derivatives of the 1/5 scale Decathlon.

C_{Z_α}	C_{Z_q}	C_{m_α}	C_{m_q}	$C_{m_{\delta_e}}$
-2.848 [4.6%]	-43.86 [5.7%]	-0.1963 [5.1%]	-3.962 [13.1%]	-0.2639 [7.4%]
C_{y_β}	C_{y_p}	C_{y_r}	C_{l_β}	$C_{l_{\delta_a}}$
-0.1561 [4.5%]	1.036 [11.7%]	0.8627 [15.2%]	-7.61e-3 [6.6%]	-1.752e-2 [16.8%]
C_{n_β}	C_{n_p}	C_{n_r}	$C_{n_{\delta_r}}$	
1.179e-2 [9.4%]	-6.768e-2 [15.2%]	-0.1322 [12.3%]	-5.691e-2 [9.1%]	

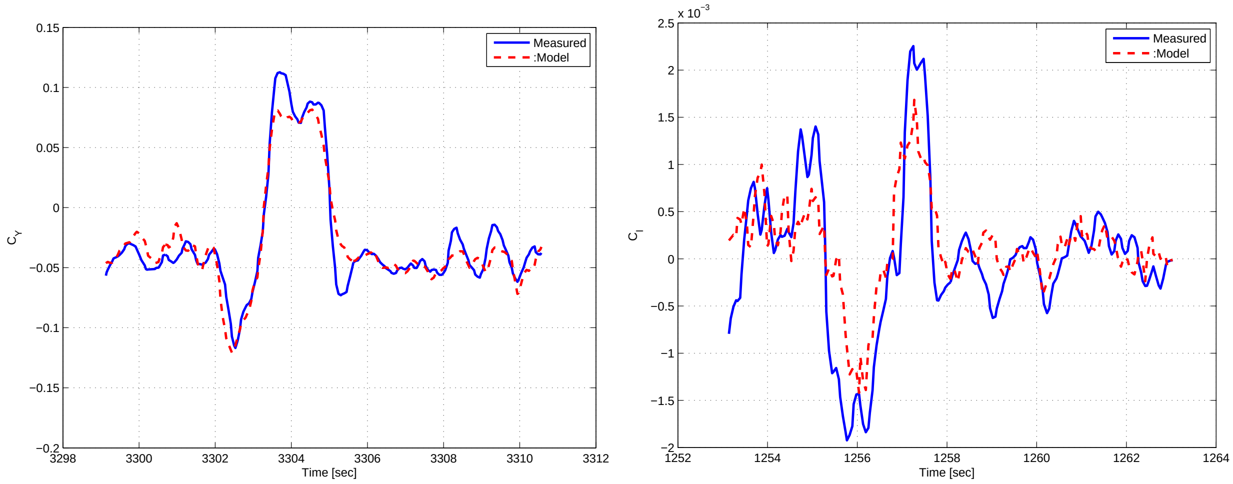
* The value inside the bracket represents the 1-σ standard deviation of the estimate error.



(a) Force coefficient C_Z

(b) Moment coefficient C_m

Figure 4. Longitudinal coefficients validation results.



(a) Force coefficient C_Y

(b) Moment coefficient C_l

Figure 5. Lateral coefficients validation results.

V. Hardware-in-the-Loop Simulation Development

This section describes the details of developing a realistic simulation environment based on Matlab/Simulink®. A complete 6-DOF nonlinear aircraft model with a linear approximation of the forces and moments is used to simulate realistic dynamic behavior of the aircraft. The nonlinear aircraft model also involves the detail modeling of subsystems such as sensors and actuators using experimental data. In addition, the external pilot command input and the flight visualization enable the simulation to be used as a virtual flight test. The actual autopilot hardware is placed inside the simulation loop in order to test and validate both the hardware and the on-board software. Four independent computer systems were used in the hardware-in-the-loop (HIL) simulation as illustrated in Fig. 6: the 6-DOF simulator, the flight visualization computer, the autopilot microcontroller, and the ground station computer console. A detailed description for this setup is given below.

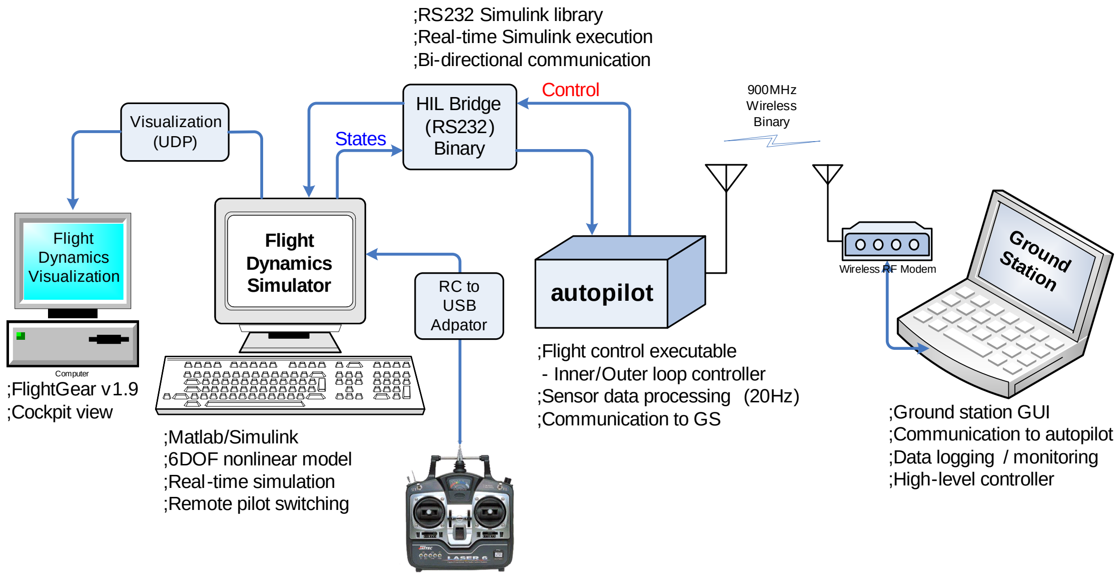


Figure 6. High fidelity hardware-in-the-loop (HIL) simulation environment.

V.A. A 6-DOF simulator

A full 6-DOF nonlinear model for the aircraft was built in the Matlab/Simulink® environment. The 12 states differential equations of motion in Eqs. (1)-(4) are utilized in conjunction with an approximation of the aerodynamic forces and moments via component buildup. The stability and control derivatives are either obtained in the manner as discussed in Sec. II.B or Sec. IV. The stability and control derivatives from the geometry of the aircraft can be chosen during the early phase of development, however, once these derivatives have been identified experimentally, the identified values are used for realistic simulation results. The 6-DOF nonlinear model becomes more realistic by employing a standard atmospheric model, an Earth and gravity model (WGS-84), and an Earth magnetic field model.[24] The output states from the simulator are processed to emulate real sensors accounting for sensor latency, random walk bias, and measurement noise. After digitized according to the word size of the micro-controller (12 bit, 4096 steps), the sensor values are transmitted to the autopilot via a serial communication. In a similar manner, the navigation states are processed for data latency and measurement noise and encapsulated in a binary GPS packet identical to the actual GPS data output. This packet is also transmitted to the autopilot at low update rate (1 Hz) via the serial communication.

In the HIL environment, the autopilot functions identically as in a real flight test, computing control commands to each control actuator. These commands are then sent back to the 6-DOF simulator in order to drive the actuator model of the control surface. Each control surface was modeled as a first order system whose time constant was identified experimentally with a rate limiter and the saturation angle. Consequently, a simulation loop is formed between the 6-DOF simulator (software) and the autopilot (hardware) exchanging the simulated data back and forth.

V.B. User interface

An R/C transmitter is connected to the 6-DOF simulator for recording remote pilot stick commands: Four channels for control surface commands ($\delta_a, \delta_e, \delta_t$ and δ_r) and two auxiliary commands for toggling between autonomous control mode and remote pilot mode. The remote pilot stick commands can override the autopilot control command at any time by switching the commands to the actuator models at user's choice. In this case, a simulation for an open loop maneuver can be conducted along the remote pilot input to validate the dynamic characteristic of the aircraft model.

In order to visualize the simulation, we adopted the use of FlighGear,[25] an open-source flight simulator visualization tool. FlightGear is a flight simulator framework, which has been widely used in various research environments.[26,27,28] Although it provides a total simulation environment in conjunction with the internal flight dynamics models, we used FlightGear as a visualization tool combining with the 6-DOF simulator already developed. By doing this, FlightGear receives the simulated states from the simulator at a fixed update rate to constantly refresh the virtual scenes that have been generated along the user's convenient view angles.

The ability to record the pilot stick command with visualization via FlightGear allows the simulation environment to replace a real experiment to save time and effort. The open loop behavior of the UAV can be tested and validated by a remote pilot via the user friendly input device, and the closed loop control performance can be verified and demonstrated by conducting virtual experiments in advance to a real flight test. Figure 7 shows a computer screen shot taken during the HIL simulation visualizing the dynamic behavior of the UAV, while displaying the corresponding state variables in the scopes.

V.C. Data communication and synchronization

The 6-DOF simulator exchanges the sensor and control command signals with the autopilot via a serial link established between the two. The sensor output data include 16 different sensor outputs from rate gyros, accelerometers, magnetometers, etc, and are packed into a custom-designed binary packet. This binary packet begins with a designated header to distinguish between different packets and ends by a trailer for data consistency check. The serial communication is configured for a baud rate of 115200 bps, which allows communicating by transmitting the sensor packet of 39 bytes at 20 Hz as well as the GPS packet of 54 bytes at 1 Hz update rate. The control command data from the autopilot are composed of four PWM commands for each actuator, resulting in a binary packet of 11 bytes in length associated with header and trailer to be transmitted at a 20 Hz update rate. The HIL bridge shown in Fig. 7 was then incorporated into the 6-DOF Simulink model to handle the bidirectional communication with a help of the serial communication block

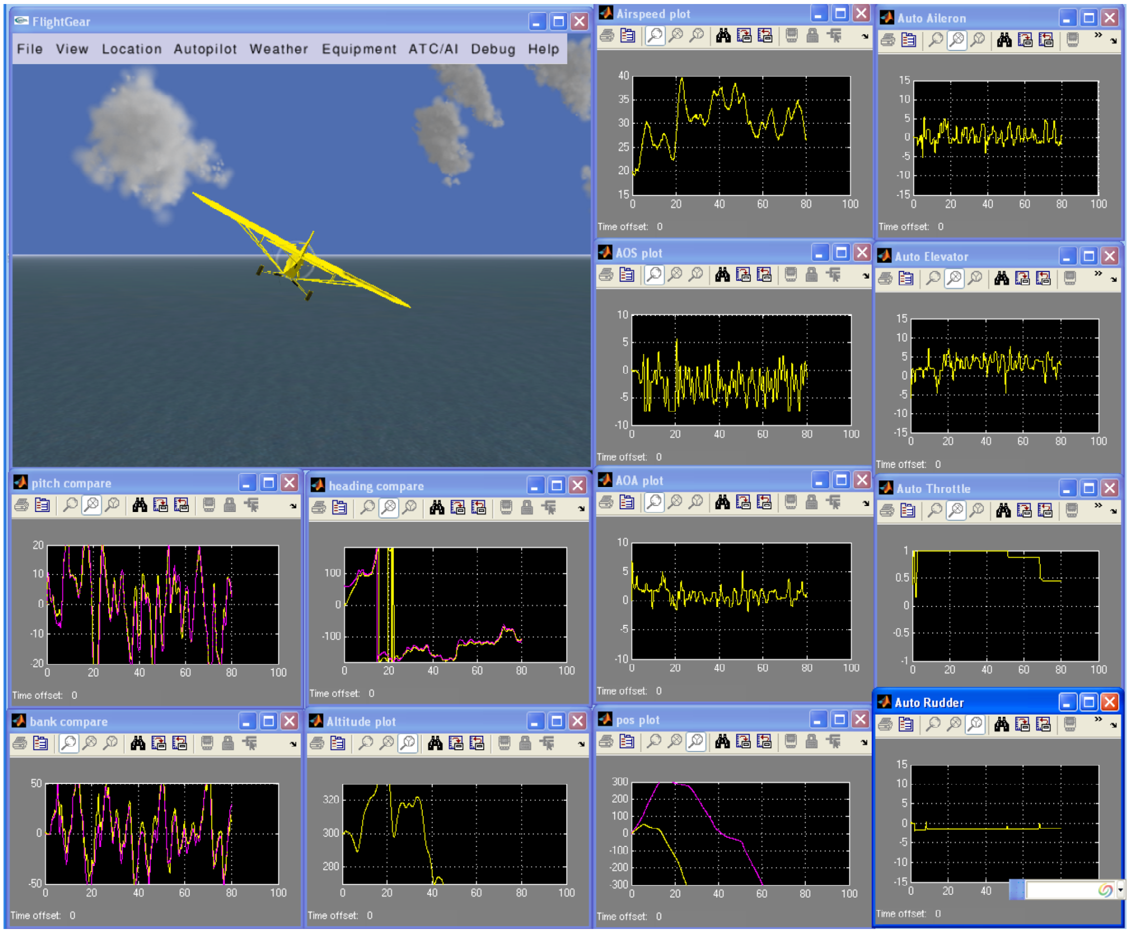


Figure 7. Hardware-in-the-loop simulation screen shot.

toolbox.[29] In addition, a dedicated S-function block[24] is utilized to send simulated output to FlightGear. The S-function block bundles the simulated data with a user defined protocol (UDP) for small size of a data packet to ensure less communication overhead over the network. Real time visualization is then made possible unless the network latency is significant. To minimize the network latency, one should locate the two hosts (for 6-DOF simulation and for 3-D visualization) within a local area network (LAN), or if the computer resources permit, one can configure to use a single computer for visualizing and executing the 6-DOF simulation on the same machine.

Because multiple systems are involved in the HIL simulation, it is all important to keep the execution timing among these systems properly synchronized during the entire simulation period for data compatibility among them. Besides, a real-time execution is necessary for the simulation to be in accord with the pilot input and the visualization at correct time step. Although the 6-DOF simulator was written in a Simulink model not running in real-time, a real-time execution was accomplished by utilizing the Realtime block set[30] in the simulation. This toolbox is based on the simple concept that if the current hardware cycle time is lower than the desired simulation step time, the block simply holds the current execution until the desired time step is triggered. For a real time simulation, the basic simulation time step of the 6-DOF simulator is carefully chosen at 100 Hz taking into consideration both the computational overhead and the faithful simulation results for accurate dynamic characteristics of the UAV. A different sampling rate at 20 Hz was utilized for the serial communication block and the FlightGear S-function block. The synchronization of the data was implicitly dealt with in a manner that the main simulation loop of the 6-DOF simulator polls each communication block at a fixed interval during the simulation, whereas the autopilot and the FlightGear receive the packets passively.

VI. Conclusion

A hardware-in-the-loop simulation environment has been built to validate hardware and software development of autopilot development of a small UAV. A full 6-dof nonlinear dynamic model has been used in conjunction with the linear approximation of the aerodynamic forces and moments. A batch parameter identification method has been used to determine the stability and control derivatives of the UAV using flight test data. Detailed models for the sensors and actuators were incorporated in the simulation along with the capability of real-time 3-D visualization and external pilot interface. The hardware-in-the-loop simulation is an indispensable tool for performing simulated flight tests in support of avionics development and validation of control laws for small UAVs with minimal cost and effort.

Acknowledgment: Partial support for this work has been provided by NSF award CMS-0510259.

References

[1] Johnson, E. N. and Fontaine, S., "Use of Flight Simulation to Complement Flight Testing of Low-cost UAVs," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Montreal, Canada, Aug. 2001, AIAA 2001-4059.

[2] Sorton, E. F. and Hammaker, S., "Simulated Flight Testing of an Autonomous Unmanned Aerial Vehicle Using FlightGear," *Infotech@Aerospace*, Arlington, VA, Sept. 2005, AIAA 2005-7083.

[3] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, John Wiley & Sons, Hobokren, NJ, 2nd ed., 2003.

[4] Etkin, B. and Reid, L. D., *Dynamics of Flight: Stability and Control*, John Wiley and Sons, New York, NY, 3rd ed., 1996.

[5] Williams, J. E. and Vukelich, S. R., "The USAF Stability and Control DATCOM," Tech. rep., McDonnell Douglas Astronautics Company, St Louis, MO, 1979, AFFDL-TR-79-3032.

[6] Watkiss, E. J., *Flight Dynamics of an Unmanned Aerial Vehicle*, Master's thesis, Naval Postgraduate School, Monterey, 1994.

[7] Blake, W. B., "Prediction of Fighter Aircraft Dynamic Derivatives using Digital Datcom," *Third Applied Aerodynamics Conference*, Colorado Springs, CO, Oct. 1985, AIAA-1985-4070.

[8] Iliff, K. W. and Taylor, L. W., "Determination of Stability Derivatives from Flight Data using a Newton-Raphson Minimization Technique," NASA Technical note, NASA, March 1972, NASA TN D-6579.

[9] Ward, D. G., Monaco, J. F., and Bodson, M., "Development and Flight Testing of a Parameter Identification Algorithm for Reconfigurable Control," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, Nov.-Dec. 1998, pp. 948–956.

[10] Shore, D. and Bodson, M., "Flight Testing of a Reconfigurable Control System on an Unmanned Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, July-Aug. 2005, pp. 698–707.

[11] Jang, J. and Tomlin, C., "Longitudinal stability augmentation system design for the DragonFly UAV using a single GPS receiver," *AIAA Guidance, Navigation, and Control Conference*, Austin, TX, 2003, pp. AIAA 2003–5592.

[12] Theodore, C. R., Tischler, M. B., and Colbourne, J. D., "Rapid Frequency-Domain Modeling Methods for Unmanned Aerial Vehicle Flight Control Applications," *Journal of Aircraft*, Vol. 41, No. 4, July-Aug. 2004, pp. 735–743.

[13] Klein, V. and Morelli, E. A., *Aircraft System Identification - Theory and Practice*, Education Series, AIAA, Blacksburg, VA, 2006.

[14] Galbraith, B., "DATCOM Predicted Aerodynamic Model," July 2004, http://www.holycows.net.

[15] Inman, D. J., *Engineering Vibration*, Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[16] Smetana, F. O., *Computer Assisted Analysis of Aircraft Performance, Stability, and Control* , McGraw-Hill, New York, 1984.

[17] Larrabee, E. E., "Practical Design of Minimum Induced Loss Propellers," *Society of Automotive Engineers, Business Aircraft Meeting and Exposition*, Wichita, KS, April 1979.

[18] Adkins, C. N. and Liebeck, R. H., "Design of Optimum Propellers," *Journal of Propulsion and Power*, Vol. 10, No. 5, 1994, pp. 676–682.

[19] Martin Hepperle, "JavaProp - Design and Analysis of Propellers," http://www.mh-aerotools.de/airfoils/javaprop.htm.

[20] Colgren, R. D. and Martin, K. E., "Flight Test Validation of Sideslip Estimation using Inertial Accelerations," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, Aug. 2000, AIAA-2000-4448.

[21] Heller, M., Myschik, S., Holzapfel, F., and Sachs, G., "Low-cost Approach based on Navigation Data for Determining Angles of Attack and Sideslip for Small Aircraft," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, TX, Aug. 2003, AIAA-2003-5777.

[22] Brown, R. G. and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises and Solutions*, John Wiley & Sons, New York, NY, 3rd ed., 1997.

[23] Jung, D. and Tsiotras, P., "Inertial Attitude and Position Reference System Development for a Small UAV," *AIAA Infotech at Aerospace*, Rohnert Park, CA, May 2007, AIAA Paper 07-2768.

[24] Unmanned Dynamics, http://www.u-dynamics.com/, *AeroSim User's Guide - Aeronautical Simulation Blockset Ver. 1.2*.

[25] "FlightGear Flight Simulator," http://www.flightgear.org/.

[26] Perry, A. R., "The FlightGear Flight Simulator," *USENIX Annual Technical Conference*, UseLinux SIG sessions, Boston, MA, June-July 2004.

[27] Shaw, A., Barnes, D., and Summers, P., "Landmark Recognition for Localisation and Navigation of Aerial Vehicles," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Oct. 2003, pp. 42– 47.

[28] Dimock, G. A., Deters, R. W., and Selig, M. S., "Icing Scenarios with the Icing Encounter Flight Simulator," *the AIAA 41st Aerospace Sciences Meeting and Exhibit*, Reno, NV, Jan. 2003, AIAA-2003-23.

[29] Leonardo Daga, "RS232 Blockset for Simulink," 2004, http://digilander.libero.it/LeoDaga/Simulink/RS232Blockset.htm.

[30] Leonardo Daga, "RT Blockset for Simulnk," 2004, http://digilander.libero.it/LeoDaga/Simulink/RTBlockset.htm.