Mechatronics 21 (2011) 789–802

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Development and application of an integrated framework for small UAV flight control development

Yew Chai Paw *, Gary J. Balas *

Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Available online 23 October 2010

Keywords:
Flight control
Unmanned Aerial Vehicle
Processor-in-the-loop
Integrated framework

ABSTRACT

This paper presents an integrated framework for small Unmanned Aerial Vehicle (UAV) flight control development. The approach provides a systematic procedure for flight control design process with a set of design tools that enables control engineers to rapidly synthesize, analyze and validate a candidate controller design. A model-based environment integrated with control synthesis, off-line and real-time simulation is developed for flight control synthesis, analysis and testings. The effectiveness of the proposed integrated framework is demonstrated by applying the framework approach to a small UAV testbed. Software-in-the-loop, processor-in-the-loop and flight testings are conducted with the synthesized controller implemented. Closed-loop performance and robustness results obtained are presented.

1. Introduction

Unmanned Aerial Vehicles (UAVs) are used worldwide today for a broad range of civil and military applications. There continues to be a growing demand for reliable and low cost UAV systems. This is especially true for small to mini-size UAV systems (less than 2 m wing span) where majority of systems are still deployed as prototypes due to their lack of reliability. Improvement in the modeling, testing and flight control for these vehicles would help to increase their reliability and performance during autonomous flight.

The traditional approach used for synthesizing, implementing and validating a flight control system in [1,2] for manned aircrafts is time consuming and resource intensive. Applying the same techniques to the small UAVs is not realistic. To reduce the cost and time to market, small UAV systems make use of low cost commercial-off-the-shelf autopilots [3]. These autopilots are often classical Proportional–integral–derivative (PID) controllers and ad-hoc methods are used to tune the controller gains in flight. This methodology is time consuming, high risk and has limitations associated with performance optimality and robustness. To shorten the development cycle and improve system reliability and robustness of the flight control system, it is important to develop an integrated framework for the flight control design process. This process would consist of a set of design tools that enables control engineers to rapidly synthesize, implement, analyze and validate a candidate controller design using iterative development cycles.

Numerous researchers have made the case for an integrated framework approach in recent years [4–9]. The central paradigm is a model-based development environment where different design tools and techniques can be formulated, deployed and applied. The different processes in model-based flight control development (shown in Fig. 1) are tightly-coupled and the development process may be severely hindered if each process is tackled as a separate problem. Hence flight control development must be looked at simultaneously in the context of dynamic modeling, control and model analysis, simulation, control design, real-time implementation, software and hardware-in-the-loop simulation and flight testing.

One of the main challenges of model-based flight control design approach is in deriving flight dynamics models with adequate fidelity to be used in different stages of controller development. If a precise validated flight dynamic model is available for the controller development, it simplifies the synthesis of a controller to achieve required performance specifications. However, mathematical models used are just an approximation of the vehicle dynamics. They are used to describe complex real flight dynamics. The result is uncertainty in predicting the actual flight dynamic responses. This problem is even more crucial in the development of small UAVs since their aerodynamic data are less well understood than full-size aircraft and limited literatures on detailed dynamics modeling are available [10]. Similarly the sensors used for measurement and control are less accurate than high end aerospace vehicle. With low cost and rapid development cycle constraints, extensive wind tunnel testings to extract aerodynamic data are usually not possible. Similarly flight test system identification approach for estimating the aerodynamic data is challenging with the low quality flight data obtained from the simple and

* Corresponding authors. Tel.: +1 612 625 6561 (Y.C. Paw).
E-mail addresses: pyewchai@dso.org.sg (Y.C. Paw), balas@aem.umn.edu (G.J. Balas).

0957-4158/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2010.09.009

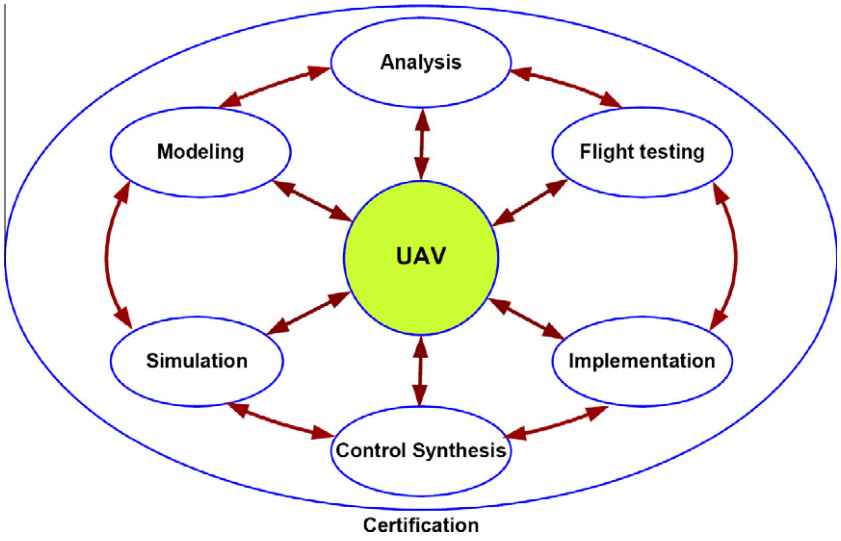


Fig. 1. Integrated framework for flight control development.

low grade sensors used onboard of the small UAVs. The application of an integrated framework will improve the fidelity of the models used through iterative updates during the flight control development cycles.

In any flight control system development, flight test validation represents the actual proof of success and assessment of whether the flight control system meets the design requirements in true environment. Flight trials are the best way to test and verify specifications though they are resource intensive and expensive. There is a need to use other validation approaches to support and augment the flight control validation process with modern flight control system becoming more complex. The ability to update and improve the accuracy of the aerodynamic and system model in achieving a high fidelity simulation model provides an attractive approach to augment the current flight control validation process. The use of simulation based testings is critical to cost reduction and time spent in the small UAVs development.

The aim of this paper is to present a systematic approach for an integrated flight control development framework that combines theoretical design tools and experimental procedures so that control engineers can easily synthesize, implement and test flight controllers on small UAV systems in a safer, cost effective and time efficient way. A commercially-off-the-shelf (COTS) radio-controlled (RC) aircraft instrumented with flight avionics system is used as a testbed to demonstrate the integrated flight control development and testing framework. Nonlinear modeling of the UAV flight dynamics is done using first principle theory in Matlab/Simulink environment with experiments carried out to determine the physical model parameters. Flight test system identification was conducted to update and verify the model developed. Parametric uncertainties derived from the experiments carried out are modeled into the nonlinear simulation model. A simplified uncertain linear lateral model is used for synthesis of a lateral-directional axis roll angle controller. The designed controller is tested in software and processor-in-the-loop integrated testing environment. The integrated framework provides an incremental and systematic approach for testing the synthesized controller before it is put onto the UAV for actual flight test.

The paper is organized as follows: In Section 2, a nonlinear simulation model of the UAV system is being developed with the aerodynamic coefficients updated using flight test system identification. Parametric uncertainties obtained from experiments are included into the nonlinear simulation model as well. Section 3 provides the flight control synthesis of the UAV roll angle controller. Details of the integrated flight control synthesis and testing framework used are covered in Section 4. Application of the integrated framework for testing the synthesized controller is illustrated in Section 5. Conclusion for the framework approach to the flight control development is given in Section 6.

2. Small UAV simulation model

The UAV simulation model is constructed in Matlab/Simulink environment through modification to the aerodynamics, propulsion and inertia AeroSim blockset [11] from Unmanned Dynamics [12]. Fig. 2 shows a simplified layout of the Aerosim 6-DOF nonlinear simulation model block diagram. Experiments are carried out to determine the required physical aircraft parameters for the simulation model. The equation of motion, earth and atmosphere blocks are not modified since they are independent of the UAV platform used. Actuator dynamics are also modeled into the simulation model to account for the actuator characteristics.

2.1. UAV platform

The COTS RC plane used has a conventional horizontal and vertical tail with rudder and elevator control surfaces (as shown in Fig. 3a). The wing has a symmetrical airfoil with aileron control surfaces. The propulsion system consists of a 600 W electric outrunner motor used to drive a 12×6 inch propeller. A summary of the UAV platform physical properties is given in Table 1.

The UAV is instrumented with a suite of avionics for the flight control development and testing. Fig. 3b shows the architecture of the avionics system. The IMU (Inertia Measurement Unit)/GPS sensor provides angular rates, linear accelerations, magnetic fields, airspeed, barometric altitude, GPS positions and velocities measurement data.

The flight computer uses *eCos* [13] real-time operating system. Sensor data are acquired into the flight computer and attitude determination is done with the acquired sensor data using a seven-state Attitude and Heading Reference Systems (AHRS) Kalman Filter [14]. At the same time, the flight computer outputs Pulse-Width Modulated (PWM) signals to drive the servo actuators and sends telemetry data information through a wireless data modem. A dual channels datalogger is used to record both raw sensor data (at 50 Hz) and flight control data (at 20 Hz). A ground control

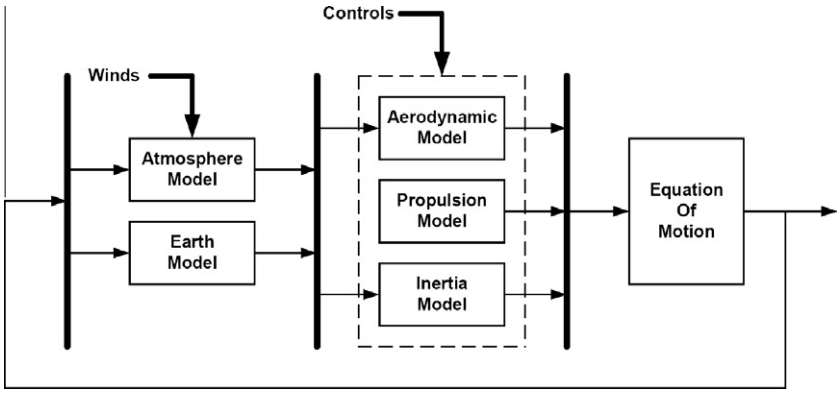
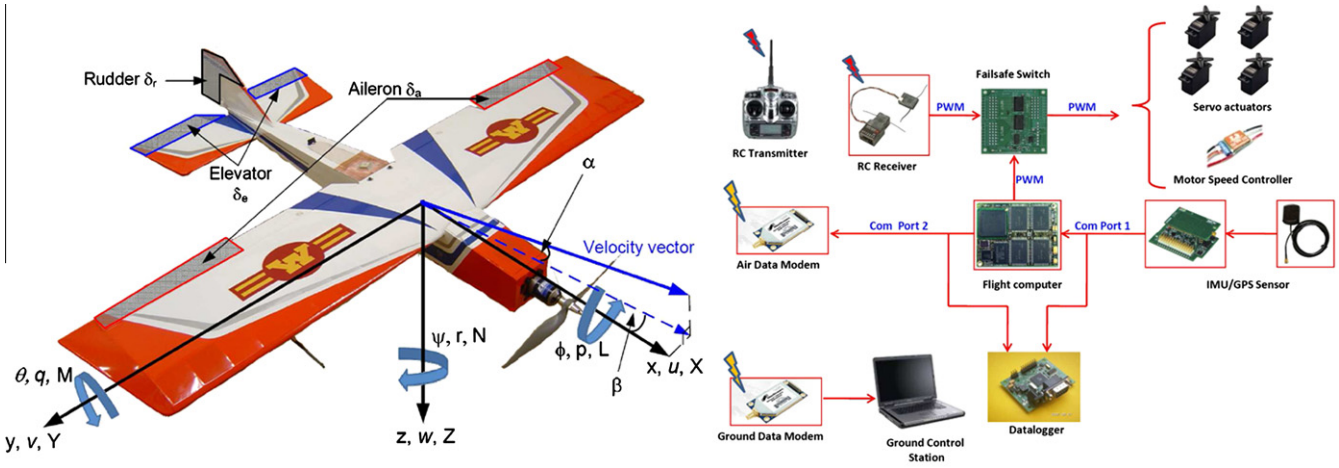


Fig. 2. Simplified layout of Aerosim 6-DOF UAV simulation model.



(a) UAV with coordinates definitions (b) Flight avionics system architecture

Fig. 3. UAV testbed.

Table 1
Summary of important aircraft geometry.

Parameter	Description	Value and units
A	Wing reference area	0.32 m^2
b	Wing span	1.2 m
$\bar{c}$	Wing chord	0.3 m
m	Gross take off weight	1.9 kg

station is used to provide real-time health monitoring of the UAV during the flight test using the telemetry data received. A failsafe switch board is used as a safety precaution to switch between flight computer commands and manual RC pilot commands.

2.2. Propulsion model

The dynamics of small UAVs with propeller propulsion system is sensitive to propulsion dynamics effects. This is due to the large propulsion system torque generated being coupled to the aircraft rigid body dynamics since the small UAV is propelled with a propeller larger relative to its aircraft size. The propulsion system models the interaction between electric motor and propeller dynamics. Applying the conservation of angular momentum, the propulsion system dynamics is described by:

$$\underbrace{(I_{mo}+I_{prop})}_{I_p}\dot{\omega}_p = T_{mo} - T_{prop} \tag{1}$$

where I_{mo} is the moment of inertia of the rotating motor body (kgm^2), I_{prop} is the moment of inertia of the propeller with spinner hub attached (kgm^2), T_{mo} is the output torque at motor shaft (Nm), T_{prop} is the torque generated by the propeller (Nm) and ω_p is the propeller angular velocity (rad/s). The motor moment of inertia I_{mo} is included because the outrunner motor has major part of the motor mass rotating with the propeller. This has a significant contribution to the total moment of inertia of the propulsion system, I_p.

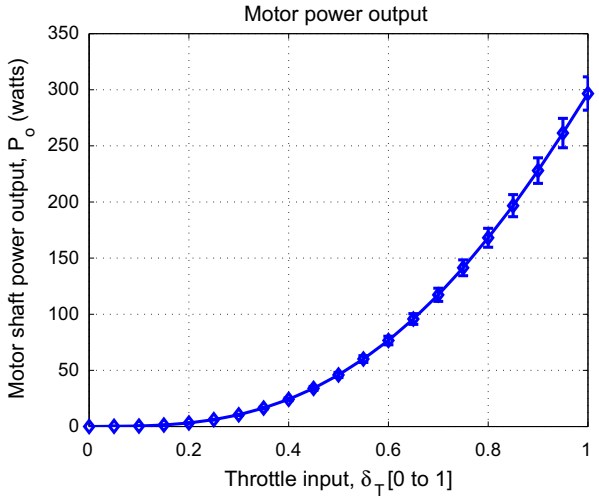
2.2.1. Propulsion motor

The propulsion motor (*E-flite Power 25 BL Outrunner Motor* [15]) performance data is not available from the manufacturer. The motor performance data is approximated using a commercial software, MotorCalc [16]. Fig. 4 shows the motor shaft output power, P_o (W), variations with the throttle stick input provided by MotorCalc. The motor shaft torque generated from the output power is:

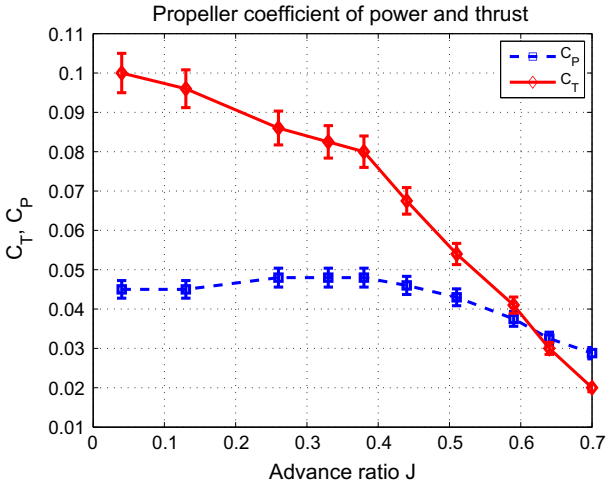
$$T_{mo} = \frac{P_o}{\omega_p} \tag{2}$$

2.2.2. Propeller characteristics

The propulsion thrust is generated by the propeller using the torque generated at motor output shaft. The propeller angular velocity depends on both the input torque available and the airspeed inflow to the propeller disk. The propeller performance is



(a) Motor shaft power output from MotorCalc



(b) Coefficient of power and thrust

Fig. 4. Propulsion system data.

characterized by advance ratio J, coefficient of thrust C_T and power C_P of the propeller [17]:

$$J = \frac{\pi V_a}{\omega_p R}, \quad C_T = \frac{F_{prop}\pi^2}{4\rho R^4 \omega_p^2}, \quad C_P = \frac{T_{prop}\pi^3}{4\rho R^5 \omega_p^2} \tag{3}$$

where ρ is the density of air (kg/m^3), R is the radius of the propeller disk (m) and V_a is the airspeed inflow into the propeller disk (m/s).

The propeller performance data is abstracted from work published in [18] on an APC 12 × 8 propeller. The thrust F_{prop} (N) and torque T_{prop} (Nm) generated from the propulsion system were measured using a force-moment sensor in a wind tunnel with different propeller angular velocities and inflow airspeeds. These data are put into lookup tables to provide the propeller thrust and torque at different airspeed conditions during the simulation. Fig. 3 shows the coefficient of thrust and power data plotted against the advance ratios from [18].

2.3. Inertia model

The inertia model contains physical geometric information of the UAV mass, center of gravity and moment of inertia coefficients. The UAV moment of inertia matrix I, with assumption that the UAV is symmetric about the xz plane, is given by:

$$I = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix} \tag{4}$$

Beside the moment of inertia matrix, the propulsion system moment of inertia coefficient, I_p, is also required. The I_{xx}, I_{yy}, I_{zz} and I_{xz} parameters in Eq. (4) are determined using compound pendulum method while the I_p parameter is determined using bifilar pendulum method. The details of the experiment setups and approaches for these two methods can be found in [19]. In each of the experiments, three sets of measurement data are collected. The largest and smallest parameter values in each of the experiments are used as the upper and lower bound values while the mean value between these two bounds is set to be the nominal value. Table 2 provides the nominal, lower and upper bound values obtained from the moment of inertia experiments.

Table 2
UAV moment of inertia data.

Moment of inertia (kgm^2)	Lower bound	Nominal	Upper bound
I_{xx}	7.74×10^{-2}	8.94×10^{-2}	1.03×10^{-1}
I_{yy}	1.24×10^{-1}	1.44×10^{-1}	1.59×10^{-1}
I_{zz}	1.34×10^{-1}	1.62×10^{-1}	1.99×10^{-1}
I_{xz}	1.12×10^{-2}	1.40×10^{-2}	1.68×10^{-2}
I_p	1.29×10^{-4}	1.30×10^{-4}	1.31×10^{-4}

2.4. Aerodynamic model

The UAV 6-DOF flight dynamic model is derived from the body axis X, Y and Z force and L, M and N moment equations [20]. Fig. 3a shows the forces and moments description in the aircraft body axis.

2.4.1. Force equations

Summation of the forces in body x, y and z axis gives linear velocity state equations [20] :

$$\dot{u} = rv - qw + \frac{\bar{q}S}{m}C_X - g\sin\theta + \frac{T}{m} \tag{5}$$

$$\dot{v} = pw - ru + \frac{\bar{q}S}{m}C_Y - g\cos\theta\sin\phi \tag{6}$$

$$\dot{w} = qu - pv + \frac{\bar{q}S}{m}C_Z - g\cos\theta\cos\phi \tag{7}$$

C_X, C_Y and C_Z are the aerodynamic force coefficients related to lift, drag and side force aerodynamic coefficients:

$$C_X = C_L\sin\alpha - C_D\cos\alpha \tag{8}$$

$$C_Z = -C_D\sin\alpha - C_L\cos\alpha \tag{9}$$

$$C_Y = C_{Y_\beta}\beta + C_{Y_{\delta r}}\delta r + \frac{b}{2V_a}(C_{Y_p}p + C_{Y_r}r) \tag{10}$$

Making small perturbation assumption to neglect higher order terms and only retains linear terms, the lift and drag coefficient, C_L and C_D, are functions of the non-dimensional coefficients given by:

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\delta e}}\delta e + \frac{c}{2V_a}(C_{L_{\dot{\alpha}}}\dot{\alpha} + C_{L_q}q) \tag{11}$$

$$C_D = C_{D_0} + C_{D_{\delta e}}\delta e + C_{D_{\delta r}}\delta r + \frac{(C_L - C_{L_{min}})}{\pi.e.AR} \tag{12}$$

The aerodynamic forces in the body x, y and z axis are given by:

$$X = \bar{q}SC_X \tag{13}$$
$$Y = \bar{q}SC_Y \tag{14}$$
$$Z = \bar{q}SC_Z \tag{15}$$

2.4.2. Moment equations

The angular rate equations obtained by taking moments about the aerodynamic center of the aircraft with non-dimensional moment coefficients c_l, c_m and c_n are given by:

$$\dot{p} - \frac{I_{xz}}{I_{xx}}\dot{r} = \frac{\bar{q}Sb}{I_{xx}}c_l - \frac{I_{zz} - I_{yy}}{I_{xx}}qr + \frac{I_{xz}}{I_{xx}}qp \tag{16}$$
$$\dot{q} = \frac{\bar{q}S\bar{c}}{I_{yy}}c_m - \frac{I_{xx} - I_{zz}}{I_{yy}}pr - \frac{I_{xz}}{I_{yy}}(p^2 - r^2) + \frac{I_p}{I_{yy}}\omega_p r \tag{17}$$
$$\dot{r} - \frac{I_{xz}}{I_{zz}}\dot{p} = \frac{\bar{q}Sb}{I_{zz}}c_n - \frac{I_{yy} - I_{xx}}{I_{zz}}pq - \frac{I_{xz}}{I_{zz}}qr - \frac{I_p}{I_{zz}}\omega_p q \tag{18}$$

where

$$c_l = c_{l_\beta}\beta + c_{l_{\delta a}}\delta a + c_{l_{\delta r}}\delta r + \frac{b}{2V_a}(c_{l_p}p + c_{l_r}r) \tag{19}$$
$$c_m = c_{m_0} + c_{m_\alpha}\alpha + c_{m_{\delta e}}\delta e + \frac{c}{2V_a}(c_{m_{\dot{\alpha}}}\dot{\alpha} + c_{m_q}q) \tag{20}$$
$$c_n = c_{n_\beta}\beta + c_{n_{\delta a}}\delta a + c_{n_{\delta r}}\delta r + \frac{b}{2V_a}(c_{n_p}p + c_{n_r}r) \tag{21}$$

The moments about the body x, y and z axis are given by:

$$L = \bar{q}Sbc_l \tag{22}$$
$$M = \bar{q}Sbc_m \tag{23}$$
$$N = \bar{q}Sbc_n \tag{24}$$

2.4.3. Kinematic equations

The kinematics of the aircraft rotation motion relating the body angular rates, Euler angles and aerodynamic angles are given by:

$$\dot{\phi} = p + tan\theta(qsin\phi + rcos\phi) \tag{25}$$
$$\dot{\theta} = qcos\phi - rsin\phi \tag{26}$$
$$\dot{\psi} = \frac{qsin\phi + rcos\phi}{cos\theta} \tag{27}$$
$$\theta = \gamma + \alpha cos\phi + \beta sin\phi \tag{28}$$

2.5. Flight test parameter identification

The aerodynamic coefficients required to model the UAV from Eqs. (11), (12), and (19)–(21) in the nonlinear simulation model are summarized in Table 3. Flight test parameter identification is used to identify the nonlinear simulation model aerodynamic coefficients at cruise flight condition. The approach taken is to estimate stability and control derivative parameters from flight data using a linear state-space model structure and subsequently converting these identified derivatives to dimensionless aerodynamic coefficients used in the nonlinear simulation model. The advantage of this approach is that it is simple to identify linear model parameters from flight test data collected using simple flight maneuvers and control input excitation signals. In this paper, identification result of the lateral dynamics coefficients $c_{l_{\delta a}}$, $c_{n_{\delta a}}$, $c_{l_{\delta r}}$, $c_{n_{\delta r}}$, c_{l_p}, c_{n_p}, c_{l_r} and c_{n_r} in the nonlinear simulation is presented to limit the scope of the paper.

Table 3
Aerodynamic coefficients required for UAV modeling.

Lift force	Drag force	Side force	Roll moment	Pitch moment	Yaw moment
C_{L_0}	C_{D_0}	C_{Y_β}	c_{l_β}	c_{m_0}	c_{n_β}
C_{L_α}	$C_{D_{\delta e}}$	$C_{Y_{\delta r}}$	$c_{l_{\delta r}}$	c_{m_α}	$c_{n_{\delta r}}$
$C_{L_{\dot{\alpha}}}$	$C_{D_{\delta r}}$	C_{Y_p}	c_{l_p}	$c_{m_{\delta e}}$	c_{n_p}
C_{L_q}		C_{Y_r}	c_{l_r}	$c_{m_{\dot{\alpha}}}$	c_{n_r}
$C_{L_{min}}$			$c_{l_{\delta a}}$	c_{m_q}	

2.5.1. Model structure for parameter identification

A linear state-space model, parameterized by physically meaningful stability and control derivatives related to the nonlinear aerodynamic coefficients, is used for the parameter identification. A parameterized two-state lateral state-space model derived from small perturbation linearization on the 6-DOF nonlinear flight dynamic model [20] is used for the parameter identification at cruise flight condition:

$$\begin{bmatrix}\dot{p}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}L_p & L_r\\ N_p & N_r\end{bmatrix}\begin{bmatrix}p\\ r\end{bmatrix} + \begin{bmatrix}L_{\delta_a} & L_{\delta_r}\\ N_{\delta_a} & N_{\delta_r}\end{bmatrix}\begin{bmatrix}\delta_a\\ \delta_r\end{bmatrix} \tag{29}$$

The control and stability derivative parameters in Eq. (29) are related to the nonlinear simulation model aerodynamic coefficients in Eqs. (19) and (21) through some dimensional quantity dependent, given in Table 5.

2.5.2. System identification flight test

Open-loop flight tests are carried out to identify the roll and yaw stability and control derivatives in Eq. (29). The avionics system onboard of the UAV (Fig. 3b) records IMU/GPS sensor and control surfaces input data at 50 Hz for the system identification. Aileron and rudder doublet control input signals are applied manually by the RC pilot from the ground in separate time window to excite the UAV open-loop lateral dynamics at a trim and wing-leveled cruise flight condition given in Table 4. Aileron doublet control input is used to perform bank-to-bank roll maneuver while rudder doublet control input is used for Dutch-roll maneuver.

2.5.3. Parameter identification

Open-loop flight test data are collected for stability and control derivative parameters identification in Eq. (29). Time-domain output error maximum likelihood parameter estimation method is used for the parameter identification due to its desirable statistical properties such as asymptotically unbiased and consistent estimates. Details of the maximum likelihood parameter identification method and its derivation can be found in [21–23]. The software toolbox, System Identification Program for Aircraft (SISPAC), developed by [23] is used for the parameter identification.

In the parameter identification setup, the state-space model in Eq. (29) provides the model structure with unknown parameters for the SISPAC program. Three sets of open-loop flight test data collected are used for the parameter identification. Table 6 provides the parameter identification results from the three data set to give model 1, model 2 and model 3.

Table 4
Desired cruise flight operating envelope.

Airspeed, V_a (m/s)	Altitude, h (m)	Throttle, δ_T (%)
16–18	90–110	45–60

Table 5
Relationships between nonlinear simulation model aerodynamic coefficients and parameter identification derivatives.

$c_{l_{\delta a}} = \frac{I_{xx}L_{\delta a}}{\bar{q}Sb}$	$c_{l_{\delta r}} = \frac{I_{xx}L_{\delta r}}{\bar{q}Sb}$	$c_{l_p} = \frac{2I_{xx}u_0L_p}{\bar{q}Sb^2}$	$c_{l_r} = \frac{2I_{xx}u_0L_r}{\bar{q}Sb^2}$
$c_{n_{\delta a}} = \frac{I_{zz}N_{\delta a}}{\bar{q}Sb}$	$c_{n_{\delta r}} = \frac{I_{zz}N_{\delta r}}{\bar{q}Sb}$	$c_{n_p} = \frac{2I_{zz}u_0N_p}{\bar{q}Sb^2}$	$c_{n_r} = \frac{2I_{zz}u_0N_r}{\bar{q}Sb^2}$

Table 6
Flight test parameter identification results.

Derivatives	Model 1	Model 2	Model 3	Nominal model
L_p	−12.0	−12.8	−11.1	−12.0
L_r	12.7	14.4	8.62	11.5
N_p	0.294	−0.448	0.687	0.120
N_r	−8.48	−6.08	−4.62	−6.55
L_{δ_a}	58.1	61.4	43.3	52.4
L_{δ_r}	13.6	12.4	8.99	11.3
N_{δ_a}	−6.58	−3.67	−4.76	−5.13
N_{δ_r}	−17.5	−15.0	−11.9	−14.7

2.6. Uncertainty modeling

The physical parameters derived from different experiments in the UAV modeling from Fig. 4, Tables 2 and 6 contain uncertainty due to imprecise nature of experimentation and limitations of physical system modeling. It is important to include these parametric uncertainties into the nonlinear simulation model so that it gives a confidence level and a bound in predicting the actual system response for controller testings using the nonlinear simulation model in the integrated framework.

2.6.1. Parametric uncertainty modeling

Parametric uncertainties are modeled in the nonlinear simulation model with each parameter described by a parametric uncertainty set using a nominal value bounded by a lower and upper bound values. An uncertain parameter P bounded within a region $[P_{lower}, P_{upper}]$ can be expressed in the form:

$$P = \overline{P}(1 + W_P\delta) \tag{30}$$

where $\overline{P}$ is the nominal parameter value, δ is any real scalar satisfying $-1 \leqslant \delta \leqslant 1$ and W_P is the uncertain gain used to scale δ to norm size of 1 given by:

$$W_P = \frac{P_{upper} - P_{lower}}{P_{upper} + P_{lower}} \tag{31}$$

The parametric uncertainties are included in the nonlinear simulation model via a multiplicative or inverse multiplicative uncertainty structure with *USS System* (Uncertain State-Space) blocks from Robust Control Toolbox [24]. This is illustrated with a simple example using the first right hand term of the angular rate equation in Eq. (16), c_l coefficient in Eq. (19) and parametric uncertainties in I_{xx}, $c_{l_{\delta a}}$, $c_{l_{\delta r}}$, c_{l_p} and c_{l_r} parameter:

$$y = \frac{\bar{q}Sb}{I_{xx}}c_l = \frac{\bar{q}Sb}{I_{xx}}[c_{l_\beta}\beta + c_{l_{\delta a}}\delta a + c_{l_{\delta r}}\delta r + \frac{b}{2V_a}(c_{l_p}p + c_{l_r}r)] \tag{32}$$

where

$$c_{l_{\delta a}} = \bar{c}_{l_{\delta a}}(1 + W_{c_{l_{\delta a}}}\delta), \quad c_{l_{\delta r}} = \bar{c}_{l_{\delta r}}(1 + W_{c_{l_{\delta r}}}\delta), \quad c_{l_p} = \bar{c}_{l_p}(1 + W_{c_{l_p}}\delta)$$

$$\frac{1}{I_{xx}} = \left(\frac{1}{\bar{I}_{xx}}\right)\left(\frac{1}{1 + W_{I_{xx}}\delta}\right), \quad c_{l_r} = \bar{c}_{l_r}(1 + W_{c_{l_r}}\delta), \quad -1 \leqslant \delta \leqslant 1$$

Each USS system block in Fig. 5 is modeled using an *ureal* [24] object with a zero nominal value and a weighted range of real value variation. All the parametric uncertainties obtained from the experiments are modeled into the nonlinear simulation model using this technique described.

2.6.2. Aerodynamic coefficients updating

The three identified lateral models in Table 6 contain variations in each of the stability and control derivatives. These identified model parameters are updated to the nonlinear simulation model using the nominal values with real parameter variations as described in Section 2.6. The nominal model is calculated using the mid-point value between the minimum and maximum value for each of the identified parameters from the three identified models. This has the benefit of reducing the size of the uncertainty bound from the nominal model.

2.6.3. Simplified uncertain linear model for controller synthesis

The two-state lateral model in Eq. (29) with eight real parametric uncertainties identified from flight test system identification is described by the nominal, lower and upper bound values for each of the stability and control derivatives given in Table 3. A simple linear uncertain model is desired for model-based controller synthesis. One approach is to overbound the real parametric uncertainty perturbation region with an unmodeled Linear Time-Invariant (LTI) dynamic uncertainty using a single complex uncertainty perturbation [25]. The computation of tight overbound from frequency domain data has been shown to reduce to a Linear Matrix Inequalities (LMI) feasibility problem in [26] where the objective is to simultaneously search for a nominal model and uncertainty weighting bound that give optimal uncertain model with the tightest bound on the data set.

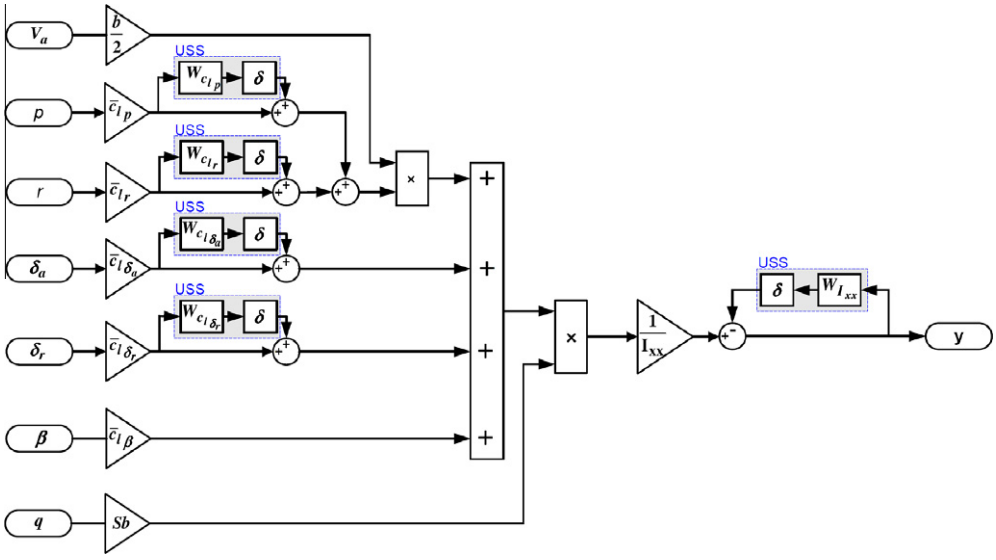


Fig. 5. Parametric uncertainty implementation example.

An input multiplicative unmodeled Linear Time-Invariant (LTI) dynamic uncertain model, G_Δ, used to cover the original set of parametric uncertain model is described by:

$$G_\Delta = G_0(I + W\Delta), \|\Delta\|_\infty \leqslant 1$$

where

- $G_0 \in \mathbb{R}^{n\times m}$ is the lateral model in Eq. (29) with nominal parameter value.
- $G_\Delta \in \mathbb{R}^{n\times m}$ has no poles on the imaginary axis.
- $W \in \mathbb{R}^{n\times n}$ is a stable and minimum phase weighting function, consisting of weights W_{ail} and W_{rud}.
- Δ is any stable transfer function with magnitude of less than or equal to 1 at each frequency point.

Fig. 6 shows the system interconnection for the input multiplicative unmodeled LTI dynamic of the uncertain lateral model. To compute the weights W_{ail} and W_{rud} required to overbound the set of real parametric uncertain lateral model, a family of 50 models is generated from the real parametric uncertain lateral model by random sampling of the parametric uncertainties between the lower and upper bound values. The objective of the sampling is to ensure the family of sampled models covers a wide range of model output responses described by the real parametric uncertain lateral model. The overbounding problem is to compute the optimal weighting function W at each frequency point using LMI feasibility problem [27] that satisfy the condition:

$$\underbrace{\{G_1, G_2, \ldots, G_{50}\}}_{\text{family of 50 sampled models}} \subseteq G_0(I + W\Delta), \quad \|\Delta\|_\infty \leqslant 1 W = \begin{bmatrix} W_{ail} & 0 \\ 0 & W_{rud} \end{bmatrix}$$

The ***ucover*** function from the Robust Control Toolbox is used to compute the overbounding weight W. This function computes an optimal frequency domain overbound and fits them to a stable, minimum phase weighting function. A second order weighting function is used to overbound these 50 sampled models and the resulting weighting functions obtained are:

$$W_{ail} = \frac{0.312s^2 + 10.7s + 33.3}{s^2 + 28.7s + 77.5},$$
$$W_{rud} = \frac{0.295s^2 + 5.08s + 40.1}{s^2 + 14.4s + 42.1} \quad (33)$$

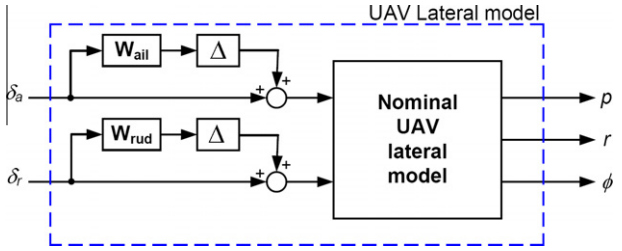


Fig. 6. Lateral UAV model with input multiplicative uncertainty.

3. Flight control architecture, synthesis and implementation

3.1. Flight control architecture

The autonomous cruise flight phase of an UAV mission involves flying at a desired cruise flight condition, performing waypoint navigation with roll angle tracking, airspeed hold, altitude hold and pitch hold mode engaged as shown in Fig. 7. The function for each of the flight modes is summarized in Table 7. Classical PID tracking controllers were synthesized and implemented on the UAV using Zigler–Nichols tuning method [19]. The scope in this paper is restricted to redesign of roll angle controller using the lateral model and model uncertainty developed with μ synthesis technique and demonstrate the integrated framework approach.

3.2. Roll angle controller synthesis

The objective of the roll angle controller is to accurately track roll angle reference commands generated by the waypoint guidance controller and to increase damping in the lateral-directional

Table 7
UAV flight modes.

Flight mode	Controller used	Function
Airspeed hold	Airspeed controller	Tracking of desired airspeed
Altitude hold	Altitude controller	Tracking of desired altitude
Pitch hold	Pitch angle controller	Tracking of reference pitch angle and provides pitch rate damping
Roll tracking	Roll angle controller	Tracking of reference roll angle and provides roll and yaw rates damping

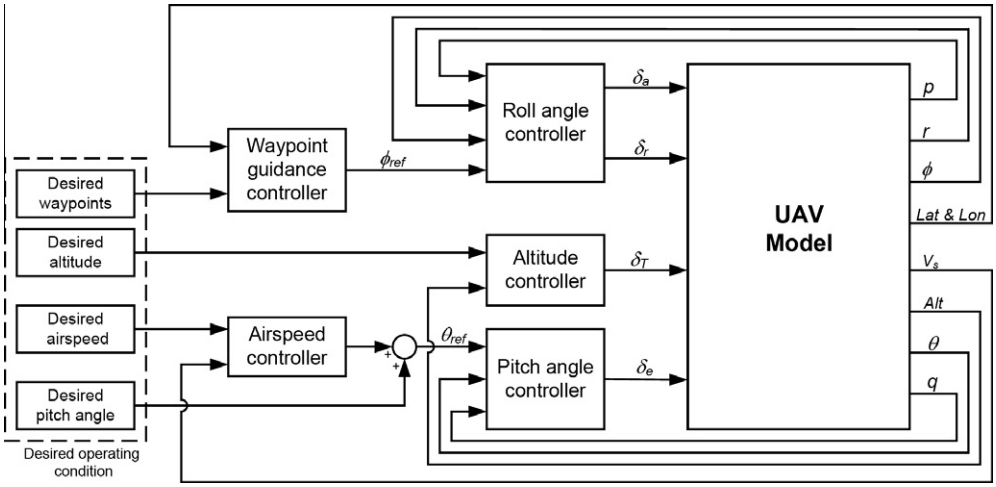


Fig. 7. UAV flight control architecture for autonomous cruise flight.

axis. These objectives are challenging because tracking of roll angle commands will cause an adverse yaw that opposes the performance objective of increased yaw rate damping. The performance criteria for the roll angle controller design are as follows:

- Track roll angle reference commands (ϕ_{ref}) with less than 6° tracking error for up to 1 rad/s bandwidth. The rise time for the roll angle tracking should be less than 2.5 s.
- The yaw rate coupling should be less than 25 deg/s across all frequencies.
- The aileron and rudder control efforts should stay within the control surface saturation limits.
- The controller should be robust and achieve desired performance objectives for the model uncertainties described in Section 2.6.

3.2.1. Weighting functions selection

The μ synthesis [28,29] controller design technique allows investigation of optimal tradeoffs between conflicting design objectives through the selection of weighting functions in the controller synthesis formulation. The weighting functions shape the frequency and magnitude content information of the input exogenous signals and output error signals, normalize and weight each of the performance requirements so that the controller synthesis problem is well-posed. This reduces the control design problem to a standard signal-based H_∞ optimization problem where the norm of the error signals is to be kept small subjected to different external signal effects. Fig. 8 shows the system interconnection with weighting functions and reference model for the roll angle controller μ synthesis.

The selection of the weighting functions and reference model in the μ synthesis are as follows:

- *Actuator model*: The actuator models, Act 1 and Act 2 (rudder and aileron servo actuators), describe the servo actuators dynamics. The outputs from the actuator models are the angular deflection rates and angle deflections. The aileron and rudder servo actuators used are the same and are modeled with first order dynamic model which mainly accounts for actuator time lag. The actuator time lag is approximated to be 20 ms. The transfer functions of the angular deflection rate and angle deflection output from the actuators are:

$$\dot{\delta}_{a,r} = \frac{50s}{s+50}, \quad \delta_{a,r} = \frac{50}{s+50}$$

- *Actuator output weighting function:* The actuator output weighting function W_{act} is used to limit the maximum angular rate and angle deflections for both the aileron and rudder control surfaces. A constant diagonal weight corresponding to the aileron angular deflection rate and angle deflection and rudder angular deflection rate and angle deflection, is used. A constant weight of 1.0 is used for the aileron and rudder angle deflections to limit the deflection angle outputs to be within ±25° while the angular deflection rates are limited to be within ±5 deg/s using a constant weighting function of 0.2. The actuator weighting function W_{act} is:

$$W_{act} = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix}$$

- *Time delay:* Time delay within the closed-loop control system is inevitable since it takes finite amount of time to process, pack, send or receive data from different hardware devices at the same time. A time delay model is used to account for the time delay (T_d) in the closed-loop system measured from the PIL simulator setup. A first order Padé approximation is used to approximate the time delay:

$$e^{-T_d s} \simeq \frac{1 - T_d s/2}{1 + T_d s/2}$$

where the time delay T_d measured from the PIL simulator setup is 0.08 s.

- *Washout filter:* Yaw damping using rudder control input during the roll angle tracking maneuver is achieved with the addition of a washout filter to the yaw rate sensor feedback path. The filter results in only high frequency yaw rate signals being fed back to the controller. The washout filter transfer function is given by:

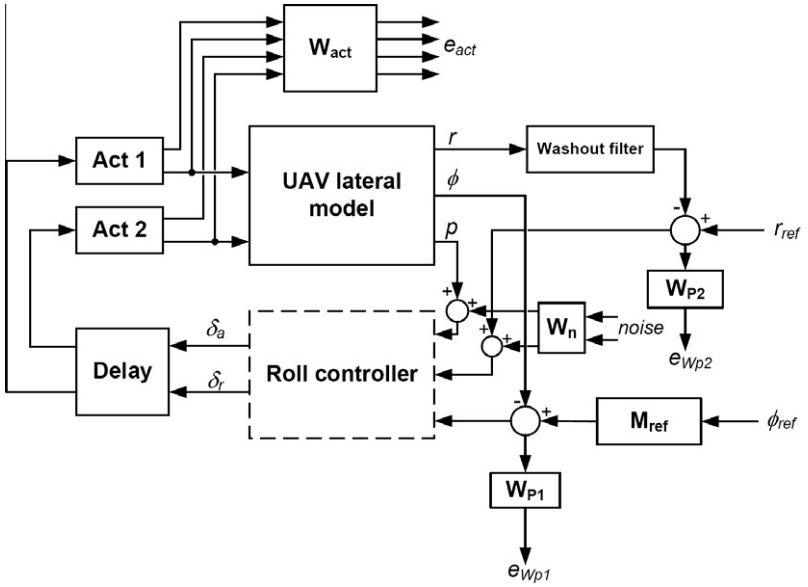


Fig. 8. System interconnection for roll angle controller μ synthesis.

$$\frac{s}{s+\omega_w}$$

where ω_w is the cutoff frequency that the yaw rate signal is attenuated such that the roll angle controller will not provide any yaw rate feedback below the cutoff frequency during steady turn. A cutoff frequency of 15 rad/s is selected.

- *Sensor noise:* A constant weight is used to model the roll and yaw gyroscopes noise in the sensor data fed back to the controller. The weight is selected based on three standard deviations bound (99.7% confidence interval) of the gyroscope noise published in [30], which is 0.03 rad/s. The sensor noise weight used is:

$$W_n = \begin{bmatrix} 0.03 & 0 \\ 0 & 0.03 \end{bmatrix}$$

- *Model reference:* A reference model is used to define the ideal closed-loop roll angle tracking response required by the UAV waypoint guidance controller that provides the roll angle reference commands. A second order reference model is used to provide a smooth roll angle reference command to the roll angle controller. A rise time of 2.2 s is chosen to have a reasonable bandwidth in the roll angle reference command. A damping ratio of 0.75 is selected to have an overshoot of less than 3% in the roll angle reference command. The second order reference model is given by:

$$M_{ref} = \frac{0.669}{s^2 + 1.227s + 0.669} \tag{34}$$

- *Performance weighting functions:* Performance weighting functions, Wp_1 and Wp_2, are used to shape the roll angle and yaw angular rate tracking error responses in achieving the required closed-loop performance specifications. Wp_1 is used to keep the mismatch between the roll angle reference command and the UAV roll angle small at low frequency with small steady-state error. The controller should roll-off at high frequency since tracking of roll angle reference command from the waypoint navigation controller is not required at high frequency. With rise time of the roll angle reference model set to 2.2 s, good tracking of the roll angle reference command signals is required between 0 and 1 rad/s. A steady-state reference roll angle tracking error of less than 6° is desired since the accuracy of roll angle attitude solution obtained from the AHRS is around ±5°. The weighting function used is:

$$Wp_1 = \frac{2.5(s+40)}{s+10} \tag{35}$$

This gives a low frequency gain of 10 for up to 1 rad/s with roll angle tracking error of less than 5.7°.Wp_2 is used to penalize the closed-loop yaw rate response with the yaw rate reference, r_{ref}, set to zero. The weighting function used is:

$$Wp_2 = \frac{1}{0.15} \tag{36}$$

This limits the yaw rate to be less than ±25 deg/s across all frequencies.

3.2.2. μ synthesis

The Matlab function ***dksyn*** in the Robust Control Toolbox is used for the μ synthesis of the roll angle controller for the system interconnection shown in Fig. 8. The roll angle controller are synthesized through minimizing of μ values using D–K iterations and solves a sequence of scaled H_∞ controller problems to achieve the robust performance for the uncertain closed-loop system. A sub-optimal μ controller is used because the order of the synthesized controller grows rapidly with D–K optimization to give high order controller which is difficult to implement in the embedded flight computer system. A roll angle controller of 31 states with γ value of 0.857 and peak μ value of 0.817 is obtained from the μ synthesis.

3.3. Controller implementation

The 31 states μ controller is reduced to a eight states controller using residualization method to preserve the low frequency range DC gain of the controller since the frequency range of interest of the roll angle controller is to have good low frequency tracking performance.

The reduced order continuous-time controller is discretized with sampling time of 0.04 s using a first-order hold method for digital implementation at 25 Hz. The first-order hold discretization method is preferred because it provides a linear interpolation between the input sampled data and does not introduce additional time delay to the discretized system unlike zero-order hold method which has a half-sample time delay [31].

4. Integrated framework

4.1. Integrated framework flight control development testing

The integrated framework for flight control development provides a systematic and progressive environment for testing the synthesized controller using the derived nonlinear simulation model containing parametric uncertainties in Section 2. Fig. 9 shows the four progressive steps used in the controller testing. In each of the test setups, different aspects and issues of the controller synthesis and implementation are tested and verified. This provides an useful tool for debugging and identifying any design or implementation issue during the development cycle. Therefore it is important to know the differences between each of the test setups to better identify the source of the problem (see Fig. 10).

4.1.1. Initial design testing

Initial design testing is the first step used to test the synthesized controller. The discrete-time controller is implemented in Simulink to validate the closed-loop performance with the nonlinear simulation model. This test verifies if the designed controller is able to meet the design requirements.

4.1.2. Software-in-the-loop testing

The second step is implementation of the controller in C code as an embedded S-function block for SIL testing in the Matlab/Simulink environment. The purpose of the SIL test is to validate the correctness and implementation of the controller C code implementation.

4.1.3. Processor-in-the-loop testing

The PIL testing is used to test the successful C code controller from SIL testing to the actual flight computer in the third step. This provides an actual test of the implemented flight control codes in the flight computer. In addition to the flight control code running on the embedded processor, other software sub-modules of the autopilot system such as AHRS attitude determination algorithm, data acquisition and telemetry communication modules are running at the same time to form a complete standalone autopilot system.

4.1.4. Flight testing

Flight test provides actual test of the complete flight control system in real flight environment. The exact flight control code that was used in the PIL testing is implemented onto the UAV flight

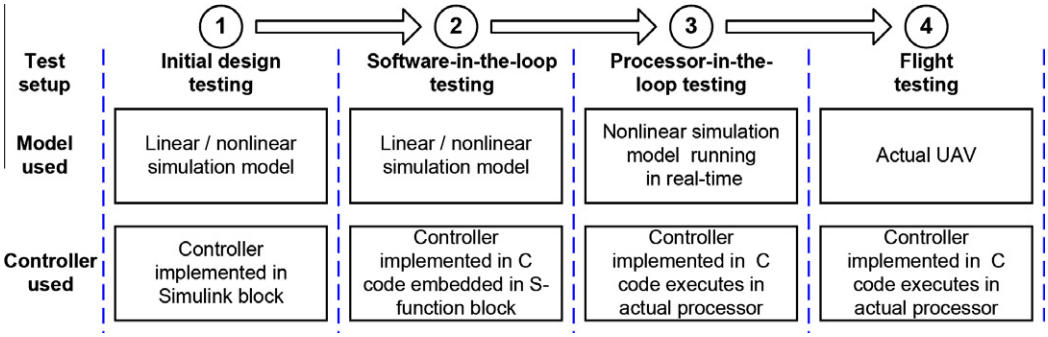


Fig. 9. Integrated framework for flight control development testing.

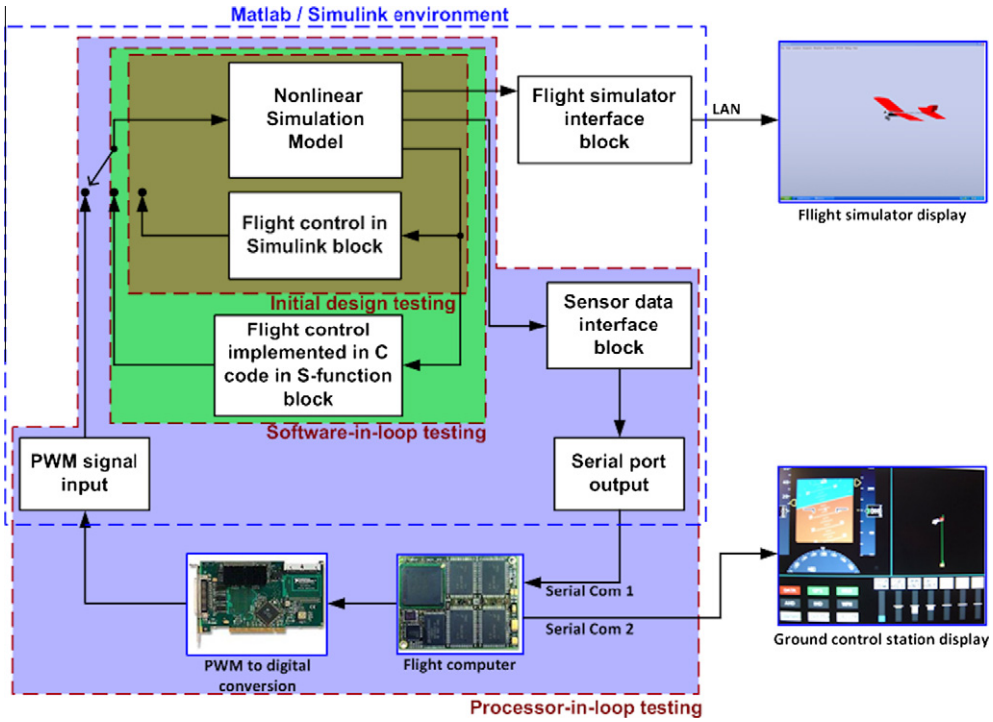


Fig. 10. Flight control development testing architecture: Initial design, software-in-the-loop and processor-in-the-loop testings.

computer system for flight testings. Data collected from the closed-loop flight tests are used to validate the nonlinear simulation model used and the performance of the flight control system implemented.

4.2. Processor-in-the-loop simulator

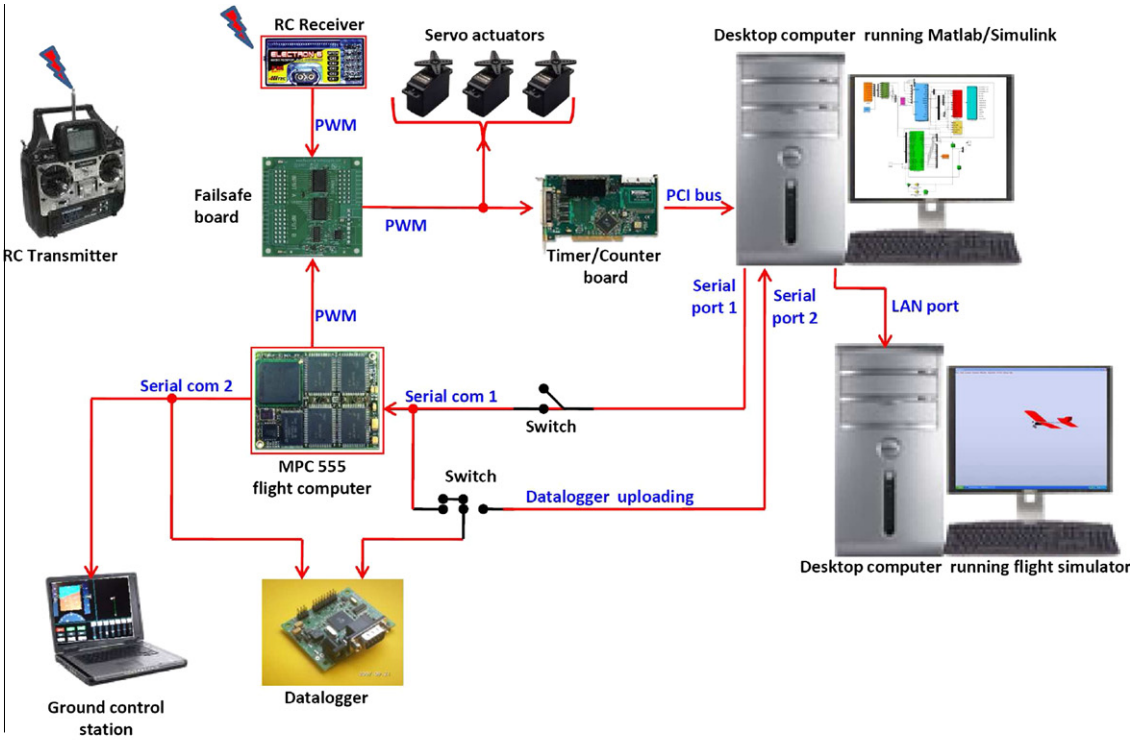
The PIL simulator setup is an extension of SIL setup that includes actual embedded flight computer with the simulation environment sending sensor data out through a communication link to the target processor that executes the embedded software code in real-time. The flight computer uses sensor data to generate control signals and sends the control signals back to the simulation model with another communication link to control the nonlinear UAV simulation model. This setup provides an intermediate step to test the synthesized controller on the actual hardware target processor before the controller is put to actual flight test.

4.2.1. PIL software system architecture

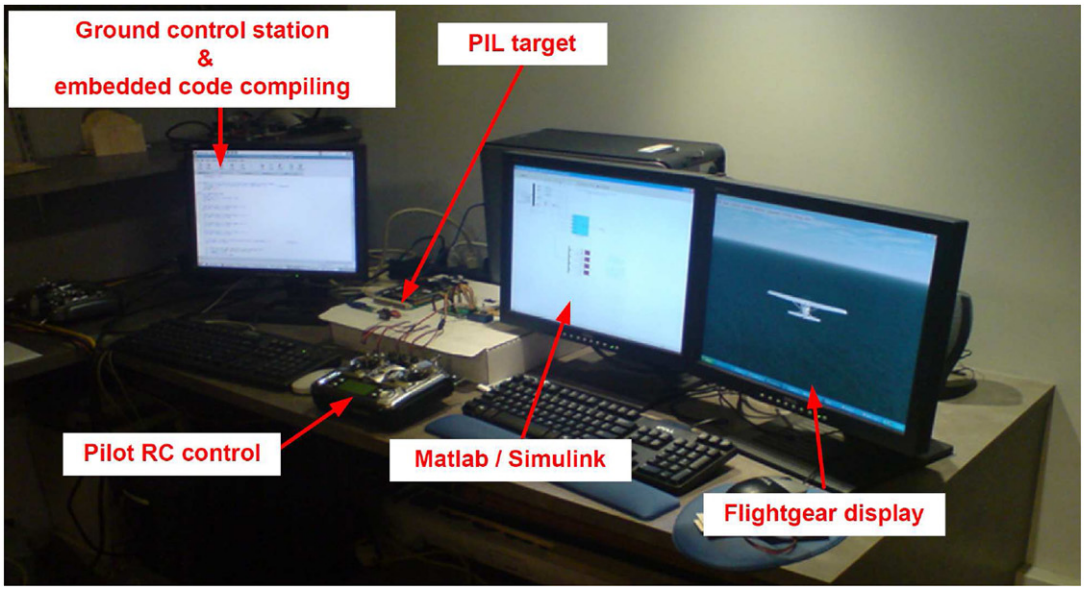
The PIL simulation model uses the same nonlinear simulation model used in the SIL testing. For the nonlinear simulation model to execute in real-time on desktop computer, Real-Time Windows Target (RTWT) toolbox [32] is used. The RTWT provides real-time execution of the generated C code to run on Windows operating system. The generated C code is able to interact with external hardware systems using the input/output (I/O) devices within the desktop computer. The entire C code generation and binary executable file are automatically generated with the RTWT toolbox.

4.2.2. PIL hardware system architecture

The PIL hardware system setup is a duplicate of the actual flight computer system on the UAV except for the IMU/GPS sensor and data modem. Fig. 11a shows the architecture of the PIL simulator hardware system setup. The desktop computer runs both the RTWT simulation and FlightGear flight simulator programs in real-time. The FlightGear flight simulator receives simulation data at 5 Hz from the RTWT simulation. The failsafe board is used to switch the PWM servo actuator control output signals between manual RC pilot or autopilot mode using the RC transmitter. This functionality allows the PIL simulator to simulate the same scenario as in the actual flight test in which the RC pilot performs manual flight before switching to autopilot mode during the flight to test the autopilot system. Fig. 11b shows the PIL simulator station.



(a) PIL simulator hardware system setup



(b) PIL simulator station

Fig. 11. PIL simulator setup.

5. Application of integrated framework and results

The integrated framework approach is used to test and validate the synthesized roll angle μ controller.

5.1. Flight test validation setup

A well-designed validation experiment is necessary to provide a realistic and consistent set of test conditions across different levels of controller testing within the integrated framework. This provides a direct and meaningful comparison for the closed-loop system performance and gives useful information for controller redesign if necessary.

5.1.1. Reference command signal design

A filtered doublet reference roll angle, ϕ_{ref}, is used to provide a smooth and piecewise continuous reference tracking signal for practical controller testing in the flight test. A second order low

pass filter, with rise time of 0.7 s and damping of 0.85, smoothes the standard doublet signal. The filter transfer function is:

$$TF_{\phi_{dbt}\to\phi_{ref}} = \frac{6.612}{s^2 + 4.371s + 6.612}$$

The amplitude of the roll angle doublet is chosen to be ±20° so that the closed-loop system stays within the controller designed operating flight envelope. The period of the doublet signal is selected to be 2.5 s. To ensure a consistent and repeatable ϕ_{ref} is being input to all the testings, the ϕ_{ref} signal profile is pre-programmed and generated by the flight computer system. Fig. 12 shows the time history of ϕ_{ref} signal designed for the validation experiment. The ϕ_{ref} signal is divided into three parts:

- $0 \leqslant t < 2$ s: The ϕ_{ref} is zero to give zero roll angle reference tracking to achieve a wing-leveled flight before the filtered doublet roll angle command is executed.
- $2 \leqslant t < 9$ s: The filtered doublet roll angle command is executed to provide dynamic reference roll angle tracking commands for the controller testing.
- $9 \leqslant t < 11$ s: The ϕ_{ref} is zero to command the UAV back to wing-leveled flight again to complete the validation experiment.

5.1.2. Validation test setup

The flight conditions and setup of the experiments have to be the same so that the controller tracking performance can be compared between different sets of validation experiments. This can be easily configured in the simulation setup for the SIL and PIL testings but not in actual flight testings since ambient environment factors such as wind gust disturbances cannot be controlled during the flight tests. These environmental disturbances are difficult to include in the SIL and PIL testings since it is not easy to measure these variables during the flight tests. Hence, precautions are taken to conduct the flight tests with minimal deviation from the flight test conditions used in the SIL and PIL testings.

In the initial design, SIL and PIL testings, the nonlinear simulation model used contains parametric uncertainties derived from experiments. The parametric uncertainties are varied repeatedly through random samplings with multiple runs (Monte-Carlo simulation runs) to test the robustness of the implemented controller in each of the test setups.

The flight conditions used for the validation experiment is the cruise flight operating condition in Table 4. The sequence of ϕ_{ref} command starts when the autopilot system is engaged by the RC pilot. In the autopilot mode, the pitch angle PID controller is engaged to provide a zero pitch angle flight condition while the roll angle controller is tracking the ϕ_{ref} command.

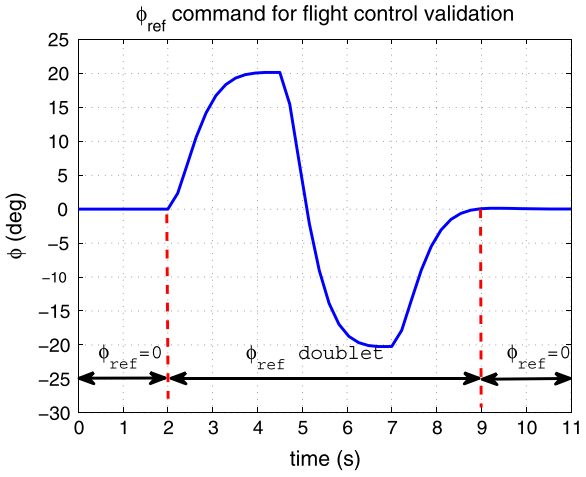


Fig. 12. Time history of ϕ_{ref} used for validation experiment.

5.2. Results

5.2.1. Initial design testing

The first level of controller testing is done using the discrete-time reduced order μ controller. Closed-loop Monte-Carlo simulation runs with model uncertainty perturbations are performed. Fig. 13a shows the ϕ angle tracking result obtained from 50 simulation runs. The result shows that the designed controller is able to track the ϕ_{ref} command well.

5.2.2. Software-in-the-loop testing

The SIL testing is performed with μ controller implemented into the Matlab S-function block. Fifty Monte-Carlo simulation runs are performed. Fig. 13b shows the result for the ϕ angle tracking. The SIL ϕ angle tracking responses obtained are almost the same as the result obtained from the initial design testing. This indicates that the controller implemented in C is correct and the implemented C code can be port to the embedded flight computer system for PIL testing.

5.2.3. Processor-in-the-loop testing

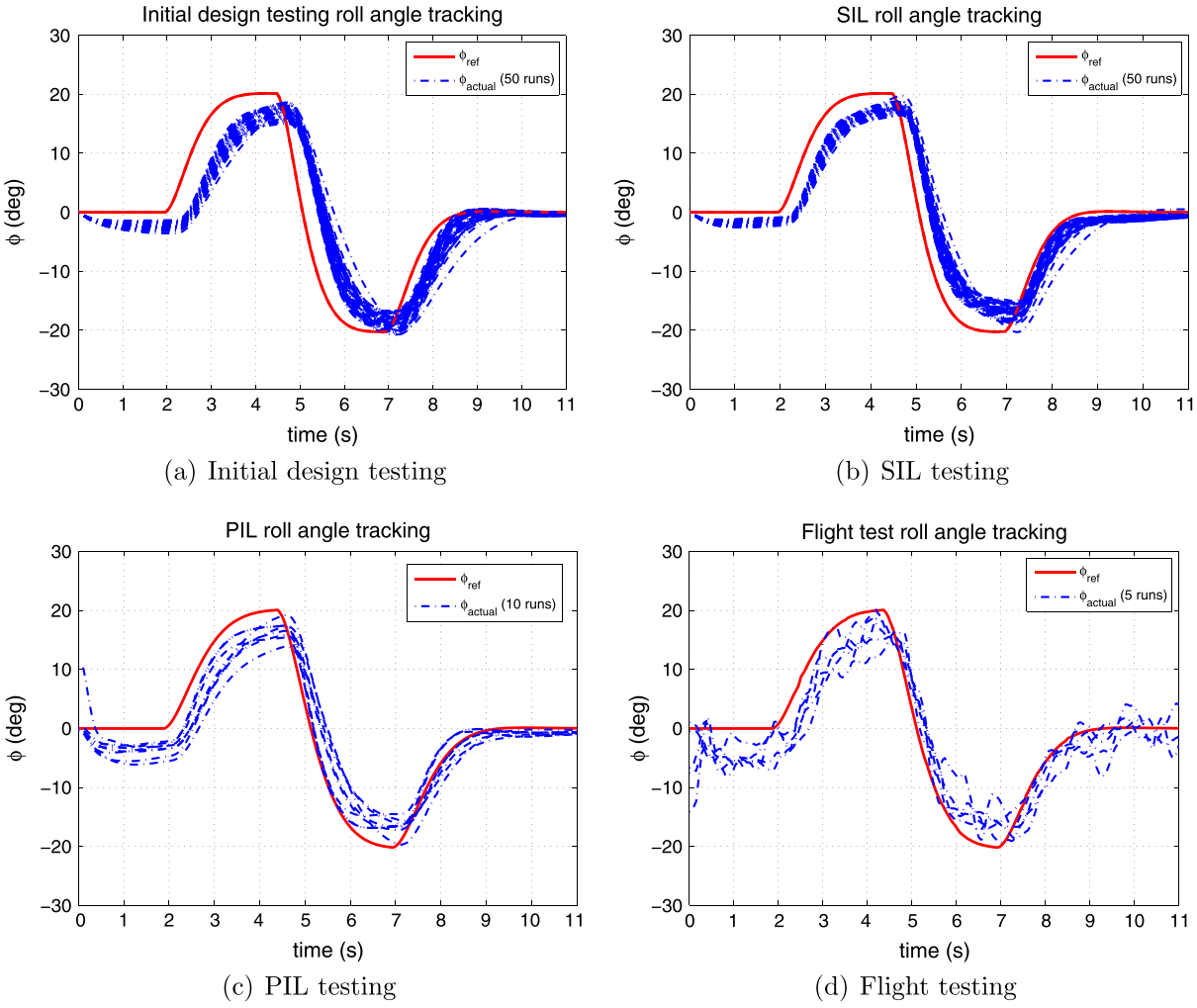
The successfully tested C code from the SIL testing was complied with the full autopilot program code and uploaded into the flight computer system for PIL testing. The PIL testing is done in the exact same manner as in flight testing where the RC pilot needs to fly the aircraft in the manual mode, trim the aircraft and switch over to autopilot mode to test the implemented controller.

Automated Monte-Carlo simulation runs used in the initial design and SIL testing cannot be used in PIL testing as the flight computer system is running the standalone autopilot routine that cannot be reset instantaneously for each different Monte-Carlo simulation run. Therefore manual variation of model uncertainty conditions was performed in each of the PIL testing run. Selective model uncertainty conditions are chosen for the PIL testing to limit the number of PIL tests. The worst-case gain analysis tools [32] is used to provide the worst-case closed-loop model uncertainty conditions for the PIL test. Ten different combinations of parameter values resulted to the worst-case gain condition within 0.5–10 rad/s frequency range are used in the nonlinear simulation model parametric uncertainties for the PIL testing. Fig. 13c shows the PIL testing result obtained using each of the worst-case gain conditions. The roll angle tracking responses obtained are similar to the result obtained from the SIL testing.

5.2.4. Flight testing

Flight testing of the UAV was conducted with the same autopilot program code tested in the PIL testing. The procedure of the flight test is carried out in the same way as the PIL testing. The RC pilot needs to fly and trim the UAV to the cruise flight condition in Table 4 before engaging the autopilot mode with the help of a ground monitoring station that provides the real-time flight test condition. For each of the runs, the reference roll command signal from Section 5.1.1 is executed by the onboard flight computer. Once the reference roll command signal execution is completed, the RC pilot assumed manual control of the UAV and trims the UAV for the next run. This procedure is repeated for four times. In each of the runs, the RC pilot trimmed the UAV towards the head wind direction to have the controller tested in a minimal cross-wind condition so that the flight test condition is similar to that used in the inital design, SIL and PIL testings.

Fig. 13d shows the flight test roll angle tracking responses obtained where $t = 0$ s (origin) is the time the autopilot was engaged. The initial roll angle values for all the runs are different since this is



(a) Initial design testing (b) SIL testing

(c) PIL testing (d) Flight testing

Fig. 13. Roll angle tracking responses.

dependent on how well the RC pilot managed to trim the UAV to wing-leveled position before the autopilot was engaged. However, regardless of the initial roll angle positions, the roll angle controller is able to have similar roll angle tracking responses obtained from the initial design, SIL and PIL testing. Repeatable and consistent tracking responses are achieved for the five different flight test runs shown in Fig. 13d. The only difference between the flight test results with the rest of the test results is that some high frequency oscillations are observed from the flight test tracking responses. This might be due to other external disturbances or high order model dynamics not captured or accounted for during the controller synthesis. Nevertheless, the flight test results still show good matching with the initial design, SIL and PIL testing. This result helps to validate:

1. Performance of the synthesized controller using the model developed achieves the control design tracking performance objectives.
2. Model and model uncertainty developed have adequate fidelity for the controller synthesis and analysis purposes.
3. Framework used for control synthesis and validation is feasible and achieves the intended objectives.

6. Conclusion

This paper presented the development of an integrated framework for flight control synthesis and validation with practical application to a small UAV testbed. The aim of this paper is to provide a systematic approach in integrating different processes in model-based flight control development using advance techniques and design tools from initial design to flight testing of the UAV system.

The 6-DOF nonlinear simulation model developed based on first principle theory was updated with flight test data to improve the fidelity of the simulation model used for controller synthesis and validation process with inclusion of experimental uncertainty modeling results. The SIL and PIL testings provide good prediction to the closed-loop tracking performance obtained from flight tests and this validates the model and model uncertainty used in the integrated framework and proves the working of the proposed integrated framework for synthesis of flight control system on the small UAV system.

References

[1] Brian SL, Lewis FL. Aircraft control and simulation. Wiley Interscience; 2003.
[2] Pratt RW. Flight control systems: practical issues in design and implementation. Institute of Engineering and Technology; 2000.
[3] Chao H, Cao Y, Chen Y. Autopilots for small fixed-wing unmanned air vehicle: a survey. In: IEEE international conference on mechatronics and automation; 2007. p. 3144–9.
[4] Jung D, Tsiotras P. Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle. AIAA Guidance, Navigation and Control Conference and Exhibit (2007-2768).
[5] Kaminer I, Yakimenko O, Dobrokhodov V, Jones K. Rapid flight test prototyping system and the fleet of UAVs and MAVs at the naval postgraduate school. In: Proceedings of the 3rd AIAA unmanned unlimited technical conference.

[6] Renfrow J, Liebler S, Denham J. F-14 flight control law design, verification and validation using computer aided engineering tools. In: Proceedings of the third IEEE conference on control applications, vol. 3; 1994. p. 359–64.
[7] Tischler M. Advances in aircraft flight control. Taylor and Francis Ltd.; 1996.
[8] Civita M, Papageorgiou G, Messner W, Kanadeo T. Design and flight testing of a high-bandwidth h_∞ loop shaping controller for a robotics helicopter. AIAA Guidance, Navigation and Control Conference and Exhibit (2002-4836).
[9] Hallberg E, Komlosy J, Rivers T, Watson M, Meeks D, Lentz IKJ. Development and application of a rapid flight test prototyping system for unmanned air vehicle. International congress on instrumentation in aerospace simulation facilities; 1999. p. 26.1–10.
[10] Jodeh NM, Blue PA, Waldron AA. Development of small unmanned aerial vehicle research platform: modeling and simulating with flight test validation. AIAA Modeling and Simulation Technologies Conference and Exhibit (2006-6261).
[11] Unmannned_Dynamics_LLC. Aerosim blockset version 1.2 user's guide; 2003.
[12] Unmanned_Dynamics. Aerosim blockset homepage. <http://www.u-dynamics.com/aerosim/default.htm>.
[13] eCos. Homepage. <http://ecos.sourceware.org/>.
[14] Jang JS, Liccardo D. Automation of small uavs using a low cost mems sensor and embedded computing platform. Tech. rep., Crossbow Technology Inc.
[15] Horizon_Hobby. Power 25 outrunner motor specifications.<http://www.horizonhobby.com/Products/Default.aspx?ProdID=EFLM4025A>.
[16] MotoCalc. Homepage. <http://www.motocalc.com/>.
[17] Kotwani K, Sane S, Arya H, Sudhakar K. Experimental characterization of propulsion system for mini aerial vehicle. In: 31st National conference on fluid mechanics and fluid power; 2004. p. 671–8.
[18] Merchant MP. Propeller performance measurement for low reynolds number unmanned aerial vehicle applications. Master's thesis, Wichita State University; 2004.
[19] Paw YC. Synthesis and validation of flight control for UAV. Ph.D. Thesis, University of Minnesota; 2009.
[20] Nelson R. Flight stability and automatic control. McGraw-Hill Science/Engineering/Math; 1997.
[21] Crassidis JL, Junkins JL. Optimal estimation of dynamic systems. Harpman & Hall/CRC; 2004.
[22] Jategaonkar RV. Flight vehicle system identification: a time domain Methodology. AIAA; 2006.
[23] Klein V, Morelli EA. Aircraft system identification: theory and practice. AIAA; 2006.
[24] Balas G, Chiang R, Packard A, Safonov M. Robust control toolbox 3 user's guide. The MathWorks; 2007.
[25] Paw YC, Balas GJ. Uncertainty modeling, analysis and robust flight control design for a small UAV system. AIAA Guidance, Navigation, and Control Conference (2008-7434).
[26] Hindi H, Seong CY, Boyd S. Computing optimal uncertainty models from frequency domain data. In: IEEE conference on decision and control, vol. 3; 2002. p. 2898–905.
[27] Balas GJ, Packard AK, Seiler PJ. Uncertain model set calculation from frequency domain data. In: Van den Hof PMJ, Heuberger PSC, Scherer CW, editors. Selected topics in model-based control: bridging rigorous theory and advanced technology. Springer-Verlag; 2009.
[28] Zhou K, Doyle JC. Essentials of robust control. Prentice Hall; 1998.
[29] Balas GJ. Robust control of flexible structures: theory and experiments. Ph.D. Thesis, California Institute of Technology; 1990.
[30] Xing Z, Gebre-Egziabher D. Modeling and bounding low cost inertial sensor errors. In: IEEE/ION position, location and navigation symposium; 2008. p. 1122–32.
[31] Ledin J. Embedded control systems in C/C++: an introduction for software developers using Matlab. CMP Books; 2004.
[32] Matlab, Simulink, real-time windows target 3 user's guide. The MathWorks Inc.; 2008.