

Research Article

Adaptive and Resilient Flight Control System for a Small Unmanned Aerial System

Gonzalo Garcia and Shahriar Keshmiri

Department of Aerospace Engineering, University of Kansas, 2120 Learned Hall 1530 W 15th Street, Lawrence, KS 66045, USA

Correspondence should be addressed to Gonzalo Garcia; garciagarretton@hotmail.com

Received 2 October 2012; Revised 18 February 2013; Accepted 18 February 2013

Academic Editor: Nicolas Avdelidis

Copyright © 2013 G. Garcia and S. Keshmiri. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The main purpose of this paper is to develop an onboard adaptive and robust flight control system that improves control, stability, and survivability of a small unmanned aerial system in off-nominal or out-of-envelope conditions. The aerodynamics of aircraft associated with hazardous and adverse onboard conditions is inherently nonlinear and unsteady. The presented flight control system improves functionalities required to adapt the flight control in the presence of aircraft model uncertainties. The fault tolerant inner loop is enhanced by an adaptive real-time artificial neural network parameter identification to monitor important changes in the aircraft's dynamics due to nonlinear and unsteady aerodynamics. The real-time artificial neural network parameter identification is done using the sliding mode learning concept and a modified version of the self-adaptive Levenberg algorithm. Numerically estimated stability and control derivatives are obtained by delta-based methods. New nonlinear guidance logic, stable in Lyapunov sense, is developed to guide the aircraft. The designed flight control system has better performance compared to a commercial off-the-shelf autopilot system in guiding and controlling an unmanned air system during a trajectory following.

1. Introduction

The effectiveness of aircraft flight control systems in off-nominal conditions is heavily dependent upon the quality and accuracy of physics-based models and numerical simulations. The accuracy of these models depends on the mathematical formulation that describes the true physics, how uncertainty affects the outputs, and whether the model mismatch can be identified and estimated. In classical flight dynamics, the measured aerodynamic forces and moments are represented by differentiable functions and expanded in Taylor series. Although it is effective in benign portions of the flight envelope, this mathematical structure is incapable of modeling nonlinear and unsteady aerodynamics associated with hazardous and adverse onboard conditions. Adverse onboard conditions include vehicle impairment; system faults, failures, and errors and vehicle damage. External hazards and disturbances include wind shear and poor visibility; wake vortices; thunderstorms; snow and icing conditions. Aircraft robust control systems are designed to deal with uncertainty in dynamic models due to modeling errors, noise, and disturbances.

The motivation of this work is to develop a robust flight control system to improve control, stability, and survivability of a small unmanned aerial system (UAS) in hazardous and abnormal flight conditions. The presented flight control system consists of guidance, navigation, and control modules with adaptive control technologies and incorporated real-time learning-based system identification using artificial neural networks (ANNs) to update the physics-based model of UAS. An extended Kalman filter (EKF) is used to improve the measurements and to estimate inaccessible variables including the inertial wind components. The research is aimed at advancing flight control systems by the integration of functionalities required to update the physics-based model and adapt the control-law in presence of model uncertainties due to unsteady and nonlinear aerodynamics. The overall configuration comprises a set of stabilizing robust H-infinity controllers in the inner loop for adjacent trim points, to be scheduled when required. New nonlinear guidance logic, stable in Lyapunov sense, is developed to guide the aircraft. The new guidance law can achieve a stable and steady-state error-free guidance through the use of integral effect and by

lifting constraints related to maximum distances and relative position with respect to the trajectory, expanding them to the entire state space.

Significant research has been carried out in areas of nonlinear guidance, navigation, adaptive control, and estimation. A broad range of research can be found on trajectory tracking, path following, and robust nonlinear controllers [1–9]. In [1], authors presented the design of guidance and control algorithms for autonomous vehicles through a simultaneous process that achieved a zero steady-state error about a trim point and, at the same time, guaranteed stability by avoiding possible cross-coupling between loops. In [2], a lateral track control-law for an autonomous aerial system that handles a larger range of wind disturbance was presented. A different approach is presented in [3] where the tracking control problem is formulated as a constrained nonlinear optimization problem. Unlike receding horizon methods, the minimization must be performed at the current instant. Reference [4] introduced nonlinear guidance logic for curved trajectory tracking reducing to a proportional-derivative control when applied to straight lines. In [5, 6], waypoint guidance was improved by including the horizontal wind estimation, allowing the aircraft to smoothly converge to a new course after switching to the next trajectory segment. A different approach for path following is presented in [7] where vector fields are used to represent desired ground track and to direct the aircraft to the desired track. More recently, in [8], a Lyapunov-based adaptive back stepping approach is used to design a flight-path controller for a nonlinear F-16 model. A nonlinear lateral guidance law, introduced in [4], is extended and modified by the authors to the longitudinal plane in [9] and tested in simulation.

Estimation of unmeasured states from noisy observations has become an essential part of flight control systems. The Kalman filter, either with an adaptive matrix gain or a fixed one, appears to be the most widely used observer (see [10–13]). Different versions of the Kalman filter ranging from linear to nonlinear, with static or time-varying Kalman gains, have been used. More refined filters such as Particle Filtering [14] and Sigma-Point [15] are designed to achieve better estimations. Another interesting approach is detailed in [16], where a nonlinear filter is designed using the state-dependent Riccati equation (SDRE) method. This technique is applicable whenever the UAS is describable in a time-varying linear format. Reference [17] estimates the external wind component in the horizontal plane to correct the velocity transformations between the respective coordinate systems. Reference [18] presents application of Kalman filters in estimation of wind components.

Different approaches using ANN for estimation of aerodynamic coefficients have been reported. In references [19–25], ANNs were propagated in a feedforward manner and their weights and biases were numerically searched by back propagating the estimation quadratic error. References [22, 23] extract the stability and control derivatives of an unstable aircraft under closed loop using a technique called Delta method. In [26–29], the fuzzy logic method was used as an alternative to ANN. The fuzzy logic modeling is used

to capture nonlinear unsteadiness aerodynamic. In [26–28], fuzzy logic was used as an alternative to ANN to capture nonlinearities and unsteadiness in aerodynamics. In [29], using postflight fuzzy logic analysis, instability, high nonlinearity, and time-dependency were found in aircraft dynamics in crosswind.

The paper is structured as follows. Section 2 presents the nonlinear modeling of the aircraft and its linear equivalent at equilibrium points. Section 3 describes the inner loop and the H-infinity controller design, techniques for gain scheduling, the outer loop and its attitude guidance logics, and assessment of guidance logic inherent stability. Section 4 describes the design of the extended Kalman filter and its main properties, and Section 5 describes the ANN and its role in aerodynamic coefficient estimation. The paper is finalized with results from comparison of the designed flight control system (FCS) with flight test data obtained from a commercial off-the-shelf (COTS) autopilot.

2. Aircraft Modeling

Following [30], a 33% scale Yak-54, shown in Figure 1(a), is modeled using Newton-Euler equations for 6-DOF rigid-body motion, as detailed in Appendix A. Its dynamics, defined in the coordinate systems shown in Figure 1(b), is a function of the states $\mathbf{x}^T(t) = [V_T, \alpha, \beta, \phi, \theta, \psi, P, Q, R, \delta_e, \delta_a, \delta_r, p_N, p_E, p_H]$, that is, airflow angles α, β , airspeed V_T , attitude angles ϕ, θ, ψ , body angular rates P, Q, R , and inertial position p_N, p_E, p_H . Throttle, elevator, aileron, and rudder servos are modeled as a first-order system with time constants $\tau_t, \tau_e, \tau_a, \tau_r$ and states $\delta_t, \delta_e, \delta_a, \delta_r$, respectively. Wind components w_N, w_E, w_H are included as uncontrollable, but measurable, disturbances to the model and will be estimated by an extended Kalman filter.

Controllable inputs to the system are propulsive and aerodynamic forces and moments, and gravity is understood as an uncontrollable input. By assuming perfect alignment with X-axis in body frame \bar{B} , propulsion or thrust $X^P(\mathbf{x}, t) = f_{X^P}(\delta_T, t)$ is modeled parametrically as a polynomial function on δ_T , as shown in Appendix B; consequently, no moments are induced. The aerodynamic forces $[\mathbf{F}_B(\mathbf{x}, t)]^A$ and moments $[\mathbf{M}_B(\mathbf{x}, t)]^A$, defined in body frame \bar{B} , are not directly measurable, and they need to be estimated from accessible data. It is customary to express them in a normalized way as a function of coefficients C_X, C_Y, C_Z and C_l, C_m, C_n , as follows:

$$\begin{aligned} [\mathbf{F}_B(\mathbf{x}, t)]^A &= \begin{bmatrix} X^A(\mathbf{x}, t) \\ Y^A(\mathbf{x}, t) \\ Z^A(\mathbf{x}, t) \end{bmatrix} = \begin{bmatrix} f_{X^A}(\mathbf{x}, t) \\ f_{Y^A}(\mathbf{x}, t) \\ f_{Z^A}(\mathbf{x}, t) \end{bmatrix} = \begin{bmatrix} \bar{q}(t) SC_X(\mathbf{x}, t) \\ \bar{q}(t) SC_Y(\mathbf{x}, t) \\ \bar{q}(t) SC_Z(\mathbf{x}, t) \end{bmatrix}, \\ [\mathbf{M}_B(\mathbf{x}, t)]^A &= \begin{bmatrix} L^A(\mathbf{x}, t) \\ M^A(\mathbf{x}, t) \\ N^A(\mathbf{x}, t) \end{bmatrix} = \begin{bmatrix} f_{L^A}(\mathbf{x}, t) \\ f_{M^A}(\mathbf{x}, t) \\ f_{N^A}(\mathbf{x}, t) \end{bmatrix} = \begin{bmatrix} \bar{q}(t) SbC_l(\mathbf{x}, t) \\ \bar{q}(t) S\bar{C}_m(\mathbf{x}, t) \\ \bar{q}(t) SbC_n(\mathbf{x}, t) \end{bmatrix}, \end{aligned} \quad (1)$$

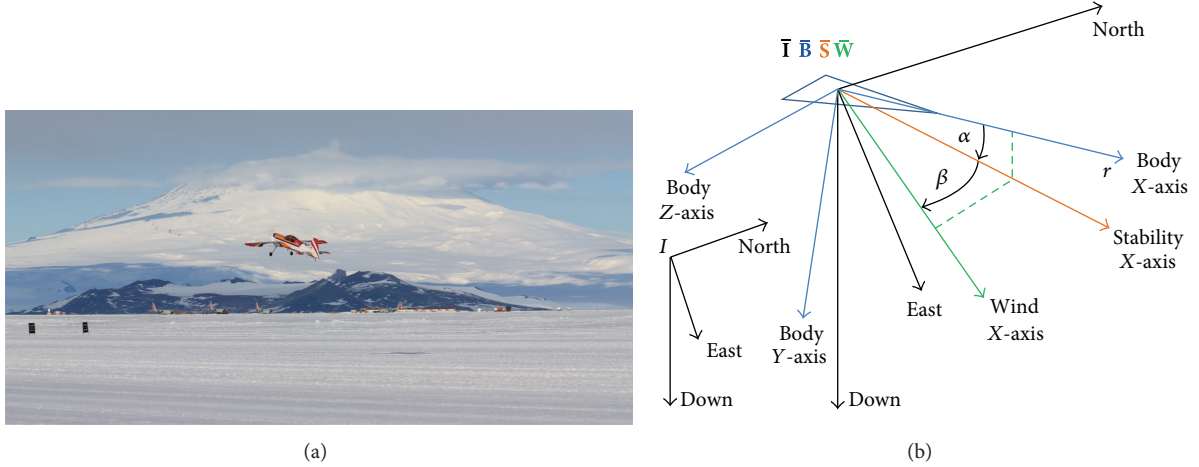


FIGURE 1: 33% Scale Yak-54 UAS flight test at Pegasus Airfield, Antarctica (2010) (a). Coordinate Systems (b).

where b and \bar{c} are the wing span and wing geometric mean chord, S is the wing reference area, and $\bar{q} = 0.5\rho V_T$ is the dynamic pressure, with air density ρ .

Although limited, a widely accepted parametric modeling is the component build-up approach, a time quasisteady and linear (in perturbed states $\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_{\text{trim}}$ around the trim condition \mathbf{x}_{trim}) model. Given its linearity and time-invariant structure, the model is valid in close proximity to the selected trim point. The present work chooses this method, based on [31], to assemble and compute these aerodynamic coefficients, as detailed in Appendix C. All constant coefficients, called stability and control derivatives, are obtained offline for selected trim conditions \mathbf{x}_{trim} . A useful tool, used to obtain stability and control derivatives in this work, is the engineering level software Advanced Aircraft Analysis (AAA) [32], developed by Design, Analysis, and Research Corporation (DARcorporation) that has a built-in aerodynamic database for different types of aircraft models. The AAA modeling is based on the chosen trim condition and the aircraft geometric parameters.

Designing a set of linear time-invariant controllers requires linear and time-invariant representations of the UAS in each equilibrium or trim point. For this purpose, a UAS nonlinear state space model derived from previous development and denoted $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \in R^n$ and $\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \in R^r$ with $\mathbf{u}(t) \in R^m$, with $\mathbf{u}^T(t) = [\delta_t^{\text{cmd}}, \delta_e^{\text{cmd}}, \delta_a^{\text{cmd}}, \delta_r^{\text{cmd}}]$ and $\mathbf{y}^T(t) = [V_T(t), \theta(t), \phi(t), \beta(t)]$, is linearized at each equilibrium $(\mathbf{x}_e^i, \mathbf{u}_e^i, \mathbf{y}_e^i)$, indexed by superscript i . Subscript e refers to the equilibrium point. The following truncated Taylor series expansion of the state space equation up to a first-order derivative, at each equilibrium point,

$$\begin{aligned} \dot{\mathbf{x}}(t) &\approx \mathbf{f}(\mathbf{x}_e^i, \mathbf{u}_e^i) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_e^i} (\mathbf{x}(t) - \mathbf{x}_e^i) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{u}_e^i} (\mathbf{u}(t) - \mathbf{u}_e^i), \\ \mathbf{y}(t) &\approx \mathbf{h}(\mathbf{x}_e^i, \mathbf{u}_e^i) + \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}_e^i} (\mathbf{x}(t) - \mathbf{x}_e^i) + \left. \frac{\partial \mathbf{h}}{\partial \mathbf{u}} \right|_{\mathbf{u}_e^i} (\mathbf{u}(t) - \mathbf{u}_e^i), \end{aligned} \quad (2)$$

gives the linear representation of the system at equilibrium. After cancellation of constant terms and defining the partial derivatives evaluated at each equilibrium point, the following linear time-invariant model, denominated \mathbf{G}^i , is obtained for control design purposes:

$$\begin{aligned} \Delta \dot{\mathbf{x}}(t) &= \mathbf{A}_G^i \Delta \mathbf{x}(t) + \mathbf{B}_G^i \Delta \mathbf{u}(t), \\ \Delta \mathbf{y}(t) &= \mathbf{C}_G^i \Delta \mathbf{x}(t) + \mathbf{D}_G^i \Delta \mathbf{u}(t). \end{aligned} \quad (3)$$

From AAA, for a straight and level flight at a constant speed of 70 (knots) with $J_x = 1.0886$ (slug \cdot ft²), $J_y = 2.1068$ (slug \cdot ft²), $J_z = 3.0382$ (slug \cdot ft²), $J_{xz} = 0.05$ (slug \cdot ft²), $m = 28$ (lb) and $g = 32.17$ (ft/sec²), [33] calculates the stability and control derivatives of a 33% scale Yak54; shown in Table 1.

The work done in [33] employed the vortex lattice model (VLM) as a second method for estimation of some of the stability and control derivatives. For uncertainty analysis, results from the AAA and VLM estimations were compared with results from flight tests system identification. Table 2 shows stability and control derivatives, the level of uncertainty (maximum and minimum), and the resultant center values and half range of each uncertainty derived from comparison. In order to switch trim conditions using gain scheduling, the same set of information and analysis had to be generated at those different trim points. In this research, and for the sake of abstraction, only information for one trim point, at 70 (knots), is presented.

At the chosen trim condition, the linear model is stable in the roll, dutch roll, phugoid, and short period modes and is unstable in the spiral mode. Figure 2 shows the linearized system poles, zeros and modes.

3. Inner and Outer Loops Design

3.1. Inner Loop. The inner loop is constituted by a set of robust H-infinity controllers and a mechanism to perform gain scheduling among them based on current dynamic conditions. The linear time-invariant plant, given in (3), is

TABLE 1: AAA stability and control derivatives for 70 knots (from [33]).

Parameter	Value	Units
C_{D0}	0.0513	[—]
$C_{D\alpha}$	0.0863	[1/rad]
$C_{D\delta e}$	0	[1/rad]
$C_{D\dot{\alpha}}$	0	[1/rad]
C_{Dq}	0	[1/rad]
C_{Du}	0.0011	[1/rad]
C_{L0}	0	[—]
$C_{L\alpha}$	4.5465	[1/rad]
$C_{L\delta e}$	0.3792	[1/rad]
$C_{L\dot{\alpha}}$	1.8918	[1/rad]
C_{Lq}	5.5046	[1/rad]
C_{Lu}	0.0017	[1/rad]
C_{m0}	0.0020	[—]
$C_{m\alpha}$	-0.3937	[1/rad]
$C_{m\delta e}$	-0.8778	[1/rad]
$C_{m\dot{\alpha}}$	-4.3787	[1/rad]
C_{mq}	-8.0532	[1/rad]
C_{mu}	0.0002	[1/rad]
$C_{y\beta}$	-0.3602	[1/rad]
$C_{y\dot{\beta}}$	0	[1/rad]
C_{yp}	0.0085	[1/rad]
C_{yr}	0.2507	[1/rad]
$C_{y\delta a}$	0	[1/rad]
$C_{y\delta r}$	0.1929	[1/rad]
$C_{l\beta}$	-0.0266	[1/rad]
$C_{l\dot{\beta}}$	0	[1/rad]
C_{lp}	-0.3819	[1/rad]
C_{lr}	0.0514	[1/rad]
$C_{l\delta a}$	0.3490	[1/rad]
$C_{l\delta r}$	0.0154	[1/rad]
$C_{n\beta}$	0.1022	[1/rad]
$C_{n\dot{\beta}}$	0	[1/rad]
C_{nr}	-0.0173	[1/rad]
C_{nr}	-0.1270	[1/rad]
$C_{n\delta a}$	-0.0088	[1/rad]
$C_{n\delta r}$	-0.0996	[1/rad]

TABLE 2: Stability and control derivatives uncertainty range knots (from [33]).

Parameter	Maximum	Minimum	Central value	Half range
C_{mq}	-4.3720	-16.1064	-10.2398	5.8666
$C_{m\delta e}$	-0.7572	-1.2289	-0.9930	0.2359
$C_{l\beta}$	-0.0220	-0.0314	-0.0267	0.0047
C_{nr}	-0.1156	-0.2890	-0.2023	0.0867
$C_{n\delta r}$	-0.0996	-0.1404	-0.1200	0.0204

augmented with design weighting matrices \mathbf{W}_1 , \mathbf{W}_2 , and \mathbf{W}_3 , as shown in Figure 3(a), with inputs $\mathbf{u}_1 \in R^{m_1}$, $\mathbf{u}_2 \in R^{m_2}$ and

outputs $\mathbf{y}_1 \in R^{p_1}$, $\mathbf{y}_2 \in R^{p_2}$, becoming the augmented plant \mathbf{P}^i .

This is done in order to harmonize otherwise conflicting performance objectives, as low frequency reference tracking and disturbance rejection, and high frequency noise cancellation, by appropriately shaping particular closed loop sensitivity functions through these frequency-dependent weighting matrices. Plant \mathbf{P}^i is then expressed as

$$\begin{aligned}\dot{\mathbf{x}}_p &= \mathbf{A}^i \mathbf{x}_p + \mathbf{B}_1^i \mathbf{u}_1 + \mathbf{B}_2^i \mathbf{u}_2, \\ \mathbf{y}_1 &= \mathbf{C}_1^i \mathbf{x}_p + \mathbf{D}_{11}^i \mathbf{u}_1 + \mathbf{D}_{12}^i \mathbf{u}_2, \\ \mathbf{y}_2 &= \mathbf{C}_2^i \mathbf{x}_p + \mathbf{D}_{21}^i \mathbf{u}_1 + \mathbf{D}_{22}^i \mathbf{u}_2,\end{aligned}\quad (4)$$

where superscript i continues to be an index among the set of augmented plants corresponding to each linearized plant \mathbf{G}^i . One of the main requirements of the chosen gain scheduling procedure is to have a homogeneous augmented plant set in terms of same structure, states, inputs, and outputs. This is achieved by having a homogeneous linearized plant set, in terms of the similar nature of zeros and poles, same weighting matrices, and same controller calculation method.

The state vector \mathbf{x}_p is constructed by adding the states of the linearized plant and the states of the weighting matrices. In particular, the inertial positions p_N, p_E, p_H are left out of the augmented state vector as they will be used in the guidance, and also yawing angle ψ is not included for being superfluous.

If the conditions detailed in Appendix D are met, then the suboptimal H_∞ control problem [34, 35] provides a controller \mathbf{K}^i that internally stabilizes the closed-loop system such that

$$\|\mathbf{T}_{\mathbf{y}_1, \mathbf{u}_1}(s)\|_\infty = \sup_{j\omega} \bar{\sigma} [\mathbf{T}_{\mathbf{y}_1, \mathbf{u}_1}(j\omega)] \leq \gamma, \quad (5)$$

for a given $\gamma > 0$, where $\mathbf{T}_{\mathbf{y}_1, \mathbf{u}_1}$ is the closed-loop transfer function from exogenous inputs \mathbf{u}_1 (including references, disturbances, and noises) to error responses \mathbf{y}_1 (including weighted error, control, and output signals). In the present work, $\mathbf{T}_{\mathbf{y}_1, \mathbf{u}_1}$ is constructed as follows:

$$\|\mathbf{T}_{\mathbf{y}_1, \mathbf{u}_1}(s)\|_\infty = \left\| \begin{bmatrix} \mathbf{W}_1(s)\mathbf{S}(s) \\ \mathbf{W}_2(s)\mathbf{R}(s) \\ \mathbf{W}_3(s)\mathbf{T}(s) \end{bmatrix} \right\|_\infty, \quad (6)$$

as a combination of desired weighted closed loop objectives, $\mathbf{S}(s)$, $\mathbf{R}(s)$, and $\mathbf{T}(s)$, that is, sensitivity, output sensitivity, and complementary sensitivity, respectively. Weighting allows their joint minimization. For an effective low frequency reference tracking and disturbance rejection, the weighting matrix \mathbf{W}_1 is built to have high singular values at the required range. For high frequency noise cancelation, \mathbf{W}_3 is designed to have high singular values in those higher frequencies. The weighting matrix \mathbf{W}_2 is designed as a constant matrix to uniformly penalize the control effort from getting unfeasible values and saturate actuators. Figure 3(b) shows singular values of the dynamic weightings chosen for the current trim condition.

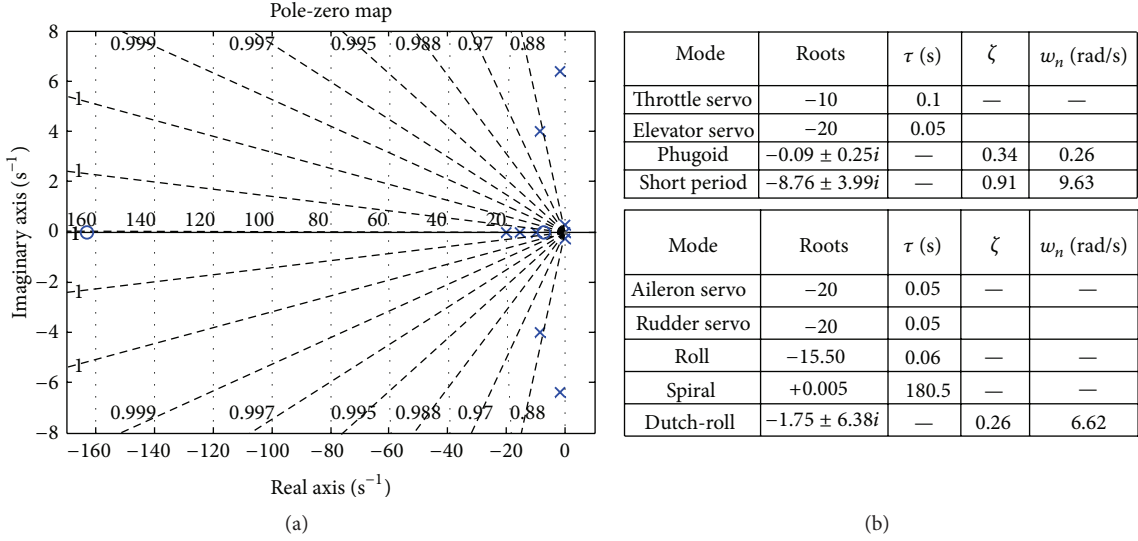
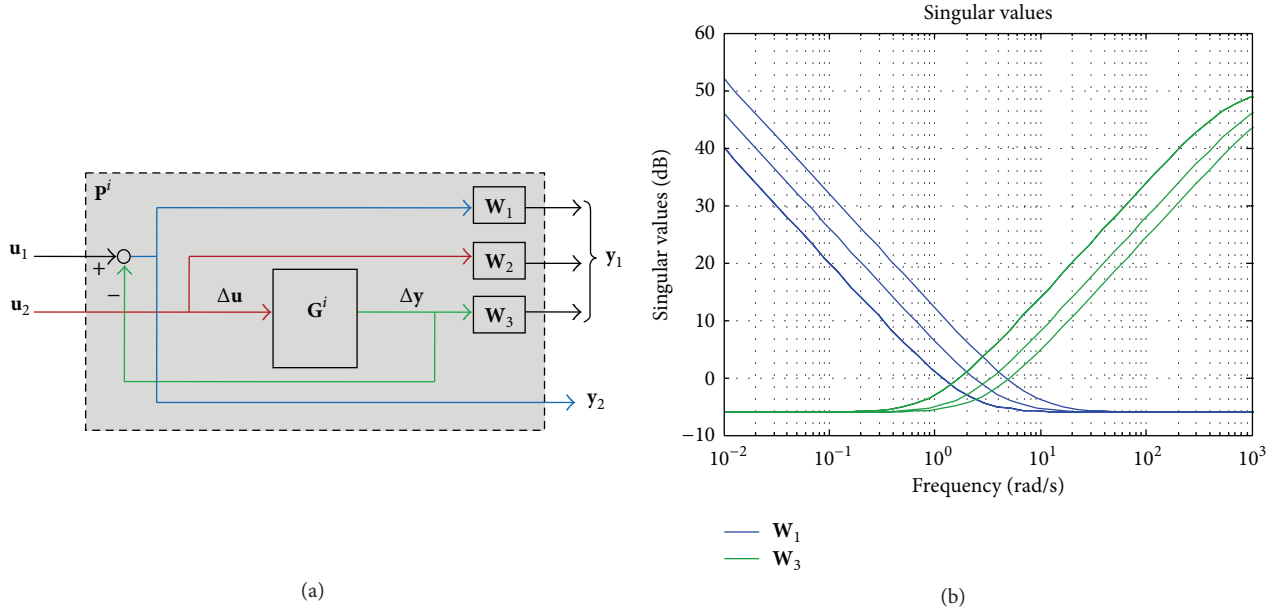


FIGURE 2: Linearized system poles: zeros (a) and modes (b).


 FIGURE 3: Augmented system (a). Singular values of weighting matrices W_1 and W_3 (b).

In order to characterize parameter uncertainty (see Table 2), a linear fractional transformation (LFT) is employed. Uncertainties are extracted and packed into Δ^i , as shown in Figure 4(a), leaving a deterministic plant \hat{P}^i , as opposed to the previous uncertain plant P^i . This LFT interconnection can be defined as

$$\begin{bmatrix} v \\ y_1 \\ y_2 \end{bmatrix} = \hat{P}^i \begin{bmatrix} \eta \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \hat{P}_{11}^i(s) & \hat{P}_{12}^i(s) & \hat{P}_{13}^i(s) \\ \hat{P}_{21}^i(s) & \hat{P}_{22}^i(s) & \hat{P}_{23}^i(s) \\ \hat{P}_{31}^i(s) & \hat{P}_{32}^i(s) & \hat{P}_{33}^i(s) \end{bmatrix} \begin{bmatrix} \eta \\ u_1 \\ u_2 \end{bmatrix}, \quad (7)$$

$$\eta = \Delta^i v, \quad u_2 = \bar{K}^i y_2,$$

or as $y_1 = F_u(F_l(\hat{P}^i, \bar{K}^i), \Delta^i)u_1 = F_l(F_u(\hat{P}^i, \Delta^i), \bar{K}^i)u_1$, where F_l corresponds to the lower LFT and F_u to the upper LFT. Assuming that $\|\Delta\|_\infty \leq 1$, then by the Small Gain Theorem, the interconnected system shown in Figure 4(a) is internally stable if and only if $\|F_l(\hat{P}^i, \bar{K}^i)\|_\infty < 1$, [34].

For each derivative given in Table 2, an uncertainty range is built considering the highest and lowest values. These extreme values are used to calculate the central value and half range value. With this information, a normalized uncertainty range is constructed and pulled out of the plant. Figure 4(b) shows the arrangement for $C_{l\beta}$ which is extensible to remaining uncertain derivatives. Rearrangement of $C_{l\beta}$ in terms of central value $C_{l\beta}^C$ and half range value $C_{l\beta}^H$ leaves its

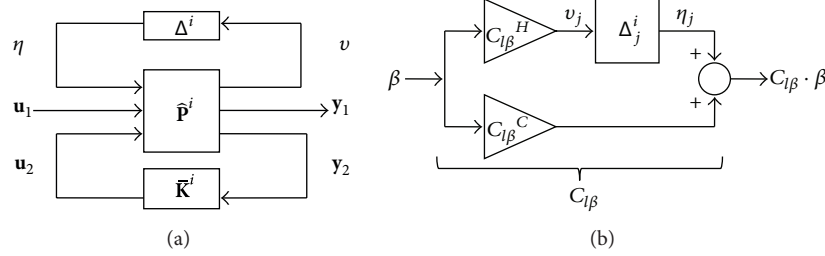


FIGURE 4: General LFT connection (a). Uncertainty normalization (b).

uncertainty block with $\|\Delta_j^i\|_\infty \leq 1$. From pulling out of the system \mathbf{P}^i each uncertainty block, an overall block Δ^i , from $\mathbf{v} = [v_1, \dots, v_j, \dots]^T$ to $\boldsymbol{\eta} = [\eta_1, \dots, \eta_j, \dots]^T$, is assembled as shown in Figure 4(a).

The 70 knots trim condition is labeled as $i = 1$, where, utilizing a set of weighting matrices as described before, a robust controller $\bar{\mathbf{K}}^1$, with $\|\mathbf{T}_{y_1, u_1}(s)\|_\infty = \gamma^1 = 0.543$, is achieved. For this closed-loop configuration, after extracting the uncertainties blocks, the infinity norms $\|\mathbf{F}_l(\hat{\mathbf{P}}^1, \bar{\mathbf{K}}^1)\|_\infty$, for each uncertain derivative, are shown in Table 3. The last row shows the worst case, the infinity norm for all uncertainties together; that is, $\|\mathbf{F}_l(\hat{\mathbf{P}}^1, \bar{\mathbf{K}}^1)\|_\infty = 0.5617$. As by construction of $\|\Delta_j^1\|_\infty \leq 1$, the Small Gain Theorem ensures that the controller robustness keeps stability facing the parameter uncertainty.

The flight control system's inner loop is equipped with a moderate fault tolerant capability. It is based on the ability to detect and identify dynamic changes due to faults that are characterized by changes in the aerodynamic forces and moment's causal functions. The implicitly assumed that static relation from aircraft states and state derivatives and controls to forces and moments is monitored through an online ANN. Changes in the network's parameters indicate unsteadiness in the aircraft dynamics, or an important trim condition transition. This information is utilized to adapt the current controller (and potentially the extended Kalman filter's model) for the new known dynamic condition, implementing a gain scheduling technique. The management of the ANN is addressed in Section 5.

Following the approach used in [36], the idea is to update the current controller $\bar{\mathbf{K}}^i$ based on an updated linear representation of the system \mathbf{G}^i , as described in (3). Its matrices $(\mathbf{A}_G^i, \mathbf{B}_G^i, \mathbf{C}_G^i, \mathbf{D}_G^i)$ are then used to compute a new robust controller through a new parameterization of \mathbf{P}^i , utilizing initially the same design parameters used in the computation of $\bar{\mathbf{K}}^i$ (i.e., same weighting matrices and same γ^i). Some simulation results are presented in Section 6. Before applying the new controller, assumptions in Appendix D should be assessed.

3.2. Outer Loop. The overall task of the inner loop, through the robust controller, is to drive the aircraft to follow external

commands generated by the outer loop, by minimizing the error between the commanded and the actual state. This minimization is performed over the error in airspeed, pitch angle, bank angle, and side slip angle; that is,

$$\begin{aligned} e_{V_T}(t) &= V_T^{\text{cmd}}(t) - V_T(t), & e_\theta(t) &= \theta^{\text{cmd}}(t) - \theta(t), \\ e_\phi(t) &= \phi^{\text{cmd}}(t) - \phi(t), & e_\beta(t) &= \beta^{\text{cmd}}(t) - \beta(t). \end{aligned} \quad (8)$$

The outer loop's task is to follow a trajectory path integrated by a set of waypoints $\{\bar{\mathbf{a}}, \bar{\mathbf{b}}, \bar{\mathbf{c}}, \dots\}$, the straight segments subtended between them $\{\mathbf{ab}, \mathbf{bc}, \mathbf{cd}, \dots\}$, and their assigned inertial speed $\{V_{\mathbf{ab}}, V_{\mathbf{bc}}, V_{\mathbf{cd}}, \dots\}$. To this end, it has two main subsystems; one is to generate the attitude commands $\phi^{\text{cmd}}, \theta^{\text{cmd}}$ based on the current inertial position p_N, p_E, p_H and speed $v_N = \dot{p}_N, v_E = \dot{p}_E, v_H = \dot{p}_H$, and the relative location of the aircraft with respect to the path to follow; a second subsystem is to generate the airspeed command V_T^{cmd} based on the path inertial speed corrected with the estimated wind. The sideslip command β^{cmd} does not require any specific logic, and it is set to zero for the entire flight.

The logic in the generation of attitude commands $\phi^{\text{cmd}}, \theta^{\text{cmd}}$, to keep the aircraft on path, is based on a non-linear guidance logic developed in [4] for lateral guidance, which for small errors and straight lines approximates a proportional-derivative controller acting on the cross-track error. It generates the ϕ^{cmd} . To calculate the θ^{cmd} , a novel extension to the longitudinal plane from [9] is developed and employed. This extension also adds new features to the original guidance idea, ensuring its stability for any location and any attitude of the aircraft. Stability in the sense of Lyapunov is proven hereinafter for the extended logic. Figure 5 shows the quantities involved for each plane separately.

The guidance logic calculates the angles η_{Lat} and η_{Lon} between the lateral and longitudinal components of the inertial velocity vector $\vec{\mathbf{V}}_{\text{Lat}}$ and $\vec{\mathbf{V}}_{\text{Lon}}$ and the corresponding components vectors $\vec{\mathbf{L}}_{\text{Lat}}$ and $\vec{\mathbf{L}}_{\text{Lon}}$ from the position vector subtended from the current UAS position $\vec{\mathbf{p}}$ and an imaginary point $\vec{\mathbf{r}}$. Point $\vec{\mathbf{r}}$ is located in the current segment between $\bar{\mathbf{a}}$ to $\bar{\mathbf{b}}$; see Figure 5.

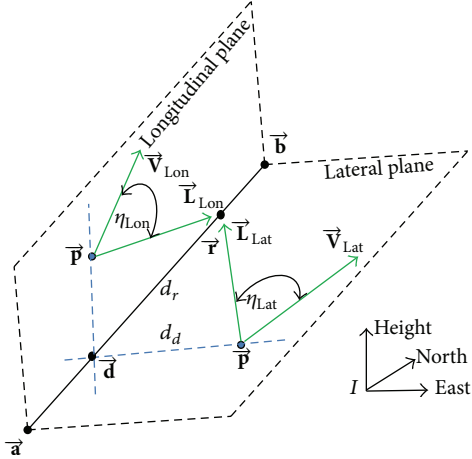


FIGURE 5: Attitude guidance logic geometry.

This point \vec{r} is also located at a fixed distance $|\vec{r} - \vec{d}| = d_r$, a numerical value which is closely dependent on the dynamic characteristics of the aircraft. The intersection between the perpendicular line from the UAS position to the current segment \vec{ab} is called \vec{d} . Point \vec{r} moves as \vec{d} moves in the direction of point \vec{b} . From the geometry seen in Figure 5, it is evident that a small or zero angle η_{Lat} or η_{Lon} is equivalent to a small or zero lateral or longitudinal cross-track error (distance between points \vec{p} and \vec{d}) and the UAS flying in the segment direction, or that the UAS is off-track, but flying toward the trajectory. Once point \vec{r} reaches a predetermined distance to the end of the segment, the segment switches to the next waypoint, now between \vec{b} to \vec{c} , and the error angles are recalculated, based on the new position vectors.

These error angles η_{Lat} and η_{Lon} (being both negative as shown in Figure 5) are translated into attitude commands through the following nonlinear relations (see [9]):

$$\begin{aligned} \phi_{cmd} &= \tan^{-1} \left\{ k_{a_{Lat}} \cdot \eta_{Lat} \cdot \left(k_{p_{Lat}} + \frac{k_{i_{Lat}}}{s} \right) \right\}, \\ \theta_{cmd} &= \tan^{-1} \left\{ k_{a_{Lon}} \cdot \eta_{Lon} \cdot \left(k_{p_{Lon}} + \frac{k_{i_{Lon}}}{s} \right) \right\}, \end{aligned} \quad (9)$$

where gains k_p and k_i are convenient gains resembling PI controllers. The integral action has been added to force the steady state angle error to zero. The arctangent function acts as a smooth limiter, avoiding unrealistic growth of the commands. The adaptive gains $k_{a_{Lat}} = 2|\vec{V}_{Lat}|^2/|\vec{L}_{Lat}|$ and $k_{a_{Lon}} = 2|\vec{V}_{Lon}|^2/|\vec{L}_{Lon}|$ are used to adequately neutralize changes in the aircraft's speed due to external disturbances, such as a wind gusts; see [4]. For the particular geometry of the nonlinear logic, errors will be zero only when both vectors are aligned and the UAS is flying with zero cross-distance or when it is pointing toward the trajectory.

As the inner loop controller also works with the airspeed state error e_{V_r} , the predefined segment inertial speed is converted from inertial \mathbf{I} to wind \mathbf{W} coordinate systems and adapted for the current wind estimation. So, when the

aircraft, driven by the controller, is reaching the commanded airspeed and is on the track, it will also be flying closer to the desired inertial speed.

3.3. Stability in the Sense of Lyapunov. In order to test the stability of the guidance law (similar to [37]), a state space representation is developed in terms of the variables $\eta_{Lat}(t)$ and $d_d(t)$, the shortest distance from the UAS to the trajectory. Then, its stability in the sense of Lyapunov is assessed. From Figure 6(a), relations are defined among guidance variables, where the time dependence has been included for clarity. Rate of change of distance $d_d(t)$ is described as follows:

$$\dot{d}_d(t) = |\vec{V}_{Lat}(t)| \sin(\eta_{Lat}(t) - \eta_{Lat}^c(t)), \quad (10)$$

where positive distance is as shown and $\eta_{Lat}^c(t)$ is defined as $\eta_{Lat}^c(t) = \tan^{-1}(d_d(t)/d_r)$. As a sign convention, angles measured counterclockwise are taken positive.

For the dynamics of $\eta_{Lat}(t)$ the centripetal acceleration $a_{cent}(t)$, shown in Figure 6(a), is defined as

$$\frac{d}{dt} (\eta_{Lat}(t) - \eta_{Lat}^c(t)) = -\frac{a_{cent}(t)}{|\vec{V}_{Lat}(t)|}. \quad (11)$$

By assuming the inner controller accomplishes a close tracking of ϕ_{cmd} , and assuming $a_{cent} = g \tan \phi$, which is valid for coordinated turns, the following dynamics holds:

$$\begin{aligned} \dot{\eta}_{Lat}(t) &= \frac{-|\vec{V}_{Lat}(t)| \eta_{Lat}(t)}{\sqrt{d_d^2(t) + d_r^2}} \\ &+ \frac{|\vec{V}_{Lat}(t)| d_r}{(d_d^2(t) + d_r^2)} \sin\left(\eta_{Lat}(t) - \tan^{-1} \frac{d_d(t)}{d_r}\right), \end{aligned} \quad (12)$$

where the speed is assumed to be constant; that is, $|\vec{V}_{Lat}(t)| = 120$ [ft/sec] ≈ 70 knots, $k_{i_{Lat}} = 0$, $k_{p_{Lat}} = 1/2g$, and $d_r = 300$ [ft]. The dynamics from (10) and (12) are used to form an autonomous nonlinear state space system and to evaluate its stability. Using a numerical and graphical solution, these equations are propagated in time for different initial conditions. Figure 6(b) shows phase portrait of the autonomous system, where all trajectories converge to the origin. It is clear that the lateral guidance logic is asymptotically stable in the sense of Lyapunov for any initial distance d_d and $-\pi > \eta_{Lat} > \pi$.

4. Extended Kalman Filter

To improve the quality of measurements and to provide estimations of unmeasured quantities, an EKF is designed based on [38] (method briefly reviewed in Appendix E) for the following dynamic model $\hat{\mathbf{x}}(t) = \hat{\mathbf{f}}(\hat{\mathbf{x}}(t), \mathbf{u}(t))$ composed of the aircraft modeling defined in Section 2. Biases and wind components are defined as new states

$$\begin{aligned}
\dot{\hat{P}}_N &= \widehat{U} \cos \widehat{\phi} \cos \widehat{\psi} + \widehat{V} (-\cos \widehat{\phi} \sin \widehat{\psi} + \sin \widehat{\phi} \sin \widehat{\theta} \cos \widehat{\psi}) + \widehat{W} (\sin \widehat{\phi} \sin \widehat{\psi} + \cos \widehat{\phi} \sin \widehat{\theta} \cos \widehat{\psi}) - \widehat{w}_N, \\
\dot{\hat{P}}_E &= \widehat{U} \cos \widehat{\theta} \sin \widehat{\psi} + \widehat{V} (\cos \widehat{\phi} \cos \widehat{\psi} + \sin \widehat{\phi} \sin \widehat{\theta} \sin \widehat{\psi}) + \widehat{W} (-\sin \widehat{\phi} \cos \widehat{\psi} + \cos \widehat{\phi} \sin \widehat{\theta} \sin \widehat{\psi}) - \widehat{w}_E, \\
\dot{\hat{P}}_H &= \widehat{U} \sin \widehat{\theta} - \widehat{V} \sin \widehat{\phi} \cos \widehat{\theta} - \widehat{W} \cos \widehat{\phi} \cos \widehat{\theta} - \widehat{w}_H, \\
\dot{\hat{V}}_T &= \frac{(\widehat{U}\dot{\widehat{U}} + \widehat{V}\dot{\widehat{V}} + \widehat{W}\dot{\widehat{W}})}{\widehat{V}_T}, \\
\dot{\hat{\alpha}} &= \frac{(\widehat{U}\dot{\widehat{W}} - \widehat{W}\dot{\widehat{U}})}{(\widehat{U}\widehat{U} + \widehat{W}\widehat{W})}, \\
\dot{\hat{\beta}} &= \frac{(\widehat{V}_T\dot{\widehat{V}} - \widehat{V}\dot{\widehat{V}}_T)}{(\widehat{V}_T^2 \cos \widehat{\beta})}, \\
\dot{\hat{\phi}} &= \widehat{P} + \tan \widehat{\theta} (\widehat{Q} \sin \widehat{\phi} + \widehat{R} \cos \widehat{\phi}), \\
\dot{\hat{\theta}} &= \widehat{Q} \cos \widehat{\phi} - \widehat{R} \sin \widehat{\phi}, \\
\dot{\hat{\psi}} &= \frac{(\widehat{R} \cos \widehat{\phi} + \widehat{Q} \sin \widehat{\phi})}{\cos \widehat{\theta}}, \\
\dot{\widehat{P}} &= (c_2 \widehat{P} + c_1 \widehat{R}) \widehat{Q} + \bar{q} S b (c_3 \widehat{C}_l + c_4 \widehat{C}_n), \\
\dot{\widehat{Q}} &= c_5 \widehat{P} \widehat{R} + c_6 (\widehat{R}^2 - \widehat{P}^2) + \bar{q} S \bar{C}_7 \widehat{C}_m, \\
\dot{\widehat{R}} &= c_8 \widehat{P} \widehat{Q} - c_2 \widehat{R} \widehat{Q} + \bar{q} S b c_4 \widehat{C}_l + \bar{q} S b c_9 \widehat{C}_n, \\
\dot{\widehat{\delta}}_T &= -10 \widehat{\delta}_T + 10 \delta_{T \text{cmd}}, \\
\dot{\widehat{\delta}}_E &= -14.30 \widehat{\delta}_E + 14.30 \delta_{E \text{cmd}}, \\
\dot{\widehat{\delta}}_A &= -14.30 \widehat{\delta}_A + 14.30 \delta_{A \text{cmd}}, \\
\dot{\widehat{\delta}}_R &= -5 \widehat{\delta}_R + 5 \delta_{R \text{cmd}}, \\
\dot{\widehat{w}}_N = \dot{\widehat{w}}_E = \dot{\widehat{w}}_H = \dot{\widehat{V}}_{Tb} = \dot{\hat{\alpha}}_b = \dot{\hat{\beta}}_b = \dot{\widehat{P}}_b = \dot{\widehat{Q}}_b = \dot{\widehat{R}}_b = \dot{\hat{a}}_{Xb} = \dot{\hat{a}}_{Yb} = \dot{\hat{a}}_{Zb} = \dot{\widehat{H}}_{Xb} = \dot{\widehat{H}}_{Yb} = \dot{\widehat{H}}_{Zb} &= 0,
\end{aligned} \tag{13}$$

where body velocities $\{\widehat{U}, \widehat{V}, \widehat{W}\}$ and their time rate of changes are defined as in Section 2 and Appendix A. The EKF model also utilizes the following output equation $\widehat{\mathbf{y}} = \widehat{\mathbf{h}}(\mathbf{x}(t), \widehat{\mathbf{u}}(t))$, designed based on available sensors including

inertial position (GPS), Euler angles (IMU), body angular rates (IMU), airflow angles vane and airspeed probe, 3D magnetic flux, body angular rate (3D gyros), translational accelerations (3D accelerometers), and elevator, aileron, and rudder deflection sensors

$$\widehat{\mathbf{h}} : \begin{cases} \widehat{y}_1 = \widehat{P}_N, & \widehat{y}_2 = \widehat{P}_E, & \widehat{y}_3 = \widehat{P}_H, & \widehat{y}_4 = \widehat{\phi}, \\ \widehat{y}_5 = \widehat{\theta}, & \widehat{y}_6 = \widehat{\psi}, & \widehat{y}_7 = \widehat{P}, & \widehat{y}_8 = \widehat{Q}, \\ \widehat{y}_9 = \widehat{R}, & \widehat{y}_{10} = \widehat{V}_T + \widehat{V}_{Tb}, & \widehat{y}_{11} = \widehat{\alpha} + \widehat{\alpha}_b, & \widehat{y}_{12} = \widehat{\beta} + \widehat{\beta}_b, \\ \widehat{y}_{13} = \widehat{H}_X + \widehat{H}_{Xb}, & \widehat{y}_{14} = \widehat{H}_Y + \widehat{H}_{Yb}, & \widehat{y}_{15} = \widehat{H}_Z + \widehat{H}_{Zb}, & \widehat{y}_{16} = \widehat{P} + \widehat{P}_b, \\ \widehat{y}_{17} = \widehat{Q} + \widehat{Q}_b, & \widehat{y}_{18} = \widehat{R} + \widehat{R}_b, & \widehat{y}_{19} = \frac{(\widehat{X}^P + \widehat{X}^A)}{m} + \widehat{a}_{Xb}, & \widehat{y}_{20} = \frac{\widehat{Y}^A}{m} + \widehat{a}_{Yb}, \\ \widehat{y}_{21} = \frac{\widehat{Z}^A}{m} + \widehat{a}_{Zb}, & \widehat{y}_{22} = \widehat{\delta}_E, & \widehat{y}_{23} = \widehat{\delta}_A, & \widehat{y}_{24} = \widehat{\delta}_R. \end{cases} \tag{14}$$

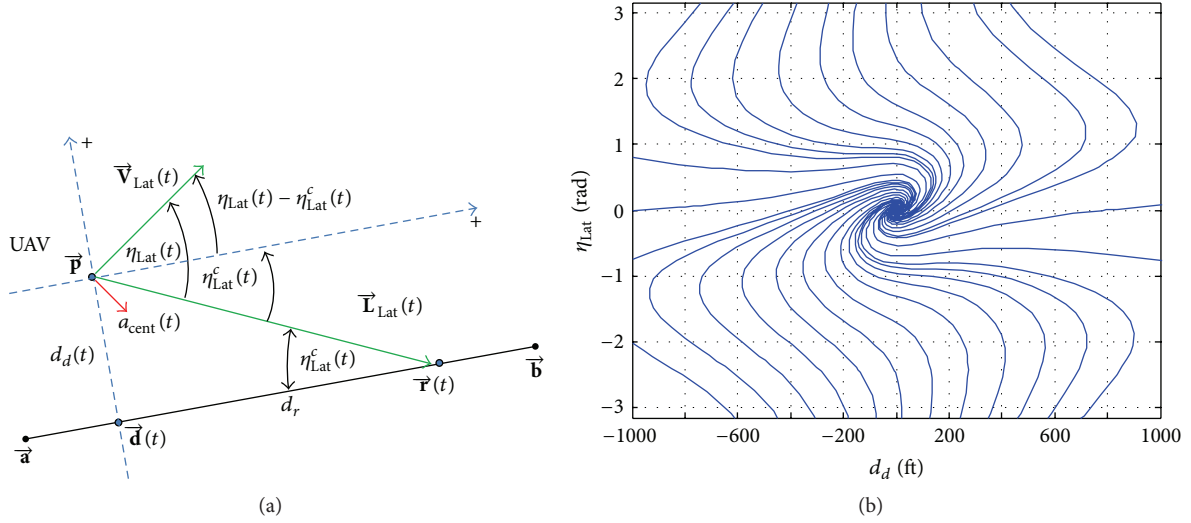


FIGURE 6: Lateral guidance geometry (a). Phase portrait of lateral guidance (b).

State vector $\hat{\mathbf{x}}[k] \in R^{\hat{n}}$, input vector $\mathbf{u}[k] \in R^{\hat{m}}$, equivalent to the system input, and measurement vector $\hat{\mathbf{y}}[k] \in R^{\hat{r}}$ are defined as

$$\begin{aligned} \hat{\mathbf{x}}[k] &= [\hat{p}_N[k], \hat{p}_E[k], \hat{p}_H[k], \hat{v}_T[k], \hat{\alpha}[k], \hat{\beta}[k], \\ &\hat{\phi}[k], \hat{\theta}[k], \hat{\psi}[k], \hat{P}[k], \hat{Q}[k], \hat{R}[k], \hat{\delta}_T[k], \hat{\delta}_E[k], \\ &\hat{\delta}_A[k], \hat{\delta}_R[k], \hat{w}_N[k], \hat{w}_E[k], \hat{w}_H[k], \hat{v}_{T_b}[k], \\ &\hat{\alpha}_b[k], \hat{\beta}_b[k], \hat{P}_b[k], \hat{Q}_b[k], \hat{R}_b[k], \hat{a}_{Xb}[k], \\ &\hat{a}_{Yb}[k], \hat{a}_{Zb}[k], \hat{H}_{Xb}[k], \hat{H}_{Yb}[k], \hat{H}_{Zb}[k]]^T, \\ \mathbf{u}[k] &= [\delta_{T_{\text{cmd}}}[k], \delta_{E_{\text{cmd}}}[k], \delta_{A_{\text{cmd}}}[k], \delta_{R_{\text{cmd}}}[k]]^T, \\ \hat{\mathbf{y}}[k] &= [\hat{p}_N[k], \hat{p}_E[k], \hat{p}_H[k], \hat{\phi}[k], \hat{\theta}[k], \hat{\psi}[k], \hat{P}[k], \\ &\hat{Q}[k], \hat{R}[k], \hat{v}_T[k] + \hat{v}_{T_b}[k], \hat{\alpha}[k] + \hat{\alpha}_b[k], \\ &\hat{\beta}[k] + \hat{\beta}_b[k], H_{X0} + \hat{H}_{Xb}[k], H_{Y0} + \hat{H}_{Yb}[k], \\ &H_{Z0} + \hat{H}_{Zb}[k], \hat{P}[k] + \hat{P}_b[k], \hat{Q}[k] + \hat{Q}_b[k], \\ &\hat{R}[k] + \hat{R}_b[k], \bar{a}_X[k] + \hat{a}_{Xb}[k], \bar{a}_Y[k] + \hat{a}_{Yb}[k], \\ &\bar{a}_Z[k] + \hat{a}_{Zb}[k], \hat{\delta}_E[k], \hat{\delta}_A[k], \hat{\delta}_R[k]]^T, \end{aligned} \quad (15)$$

where the subscript b refers to bias, a refers to acceleration, and \hat{H} indicates magnetic flux. Subscripts $\{X, Y, Z\}$ are defined in body frame $\bar{\mathbf{B}}$ and the subscript zero denotes the inertial reference magnetic flux for the flying area, assumed

fixed. Variable \bar{a} is defined as a function of estimated states, as follows:

$$\begin{bmatrix} \bar{a}_x \\ \bar{a}_y \\ \bar{a}_z \end{bmatrix} = \begin{bmatrix} \hat{R}\hat{V} - \hat{Q}\hat{W} + \frac{(\hat{T} + \hat{q}\hat{S}\hat{C}_X)}{m} \\ -\hat{R}\hat{U} + \hat{P}\hat{W} + \frac{\hat{q}\hat{S}\hat{C}_Y}{m} \\ \hat{Q}\hat{U} - \hat{P}\hat{V} + \frac{\hat{q}\hat{S}\hat{C}_Z}{m} \end{bmatrix}, \quad (16)$$

where \hat{q} is the estimated dynamic pressure and $\{\hat{C}_X, \hat{C}_Y, \hat{C}_Z\}$ are the estimated body aerodynamic force coefficients. The calculated accelerations $\{\bar{a}_x, \bar{a}_y, \bar{a}_z\}$ plus the estimated biases $\{\hat{a}_{Xb}, \hat{a}_{Yb}, \hat{a}_{Zb}\}$, as shown in (15), are used as the estimated accelerations. They do not include gravity effects to keep consistency to accelerometer measurements. As in Section 2 and Appendix A, the same relations apply to the estimated stability and body aerodynamic coefficients and to the estimated thrust \hat{T} . The body magnetic flux $\{H_X, H_Y, H_Z\}$ is related to the inertial magnetic flux $\{H_{N0}, H_{E0}, H_{D0}\}$ as follows:

$$\begin{bmatrix} H_X \\ H_Y \\ H_Z \end{bmatrix} = C_I^B(\phi, \theta, \psi) \begin{bmatrix} H_{N0} \\ H_{E0} \\ H_{D0} \end{bmatrix} + \begin{bmatrix} \hat{H}_{Xb} \\ \hat{H}_{Yb} \\ \hat{H}_{Zb} \end{bmatrix}, \quad (17)$$

where C_I^B corresponds to the direction cosine matrix for rotation from inertial reference \mathbf{I} to body reference $\bar{\mathbf{B}}$. The inertial magnetic flux $\{H_{N0}, H_{E0}, H_{D0}\}$ is a known fixed data obtained for the area to be flown.

5. Artificial Neural Network

A feedforward ANN with a single hidden layer, containing a finite number of neuron cells with sigmoid-type activation

functions and a linear activation functions at the output cells, intrinsically materializes the universal approximation theorem. It approximates any multivariate continuous function on a compact domain subset of R^m to any degree of accuracy. Based on this theorem, proven originally in [39], for any given continuous function $f_j(\mathbf{X}) \in R^1$, with $\mathbf{X} = [x_1, x_2, \dots, x_n]^T \in R^n$, both normalized within the range $[-1, 1]$ and for $\delta > 0$, there exist an integer N and a set of parameters (weights and biases) $b_i, c_j \in R$, and $\mathbf{w}_i \in R^n$ and $a_{ji} \in R$ with $i = 1 \dots N$ and j a fixed value from the set $\{1 \dots p\}$ indicating a particular function within a set of related functions, as is the case of aircraft accelerations, such that

$$F_j(\mathbf{X}) = \sum_{i=1}^N a_{ji} \varphi(\mathbf{w}_i^T \mathbf{X} + b_i) + c_j \quad (18)$$

$$\widehat{\mathbf{X}}^T[k] = \begin{bmatrix} \widehat{V}_T[k - N_A + 1] & \widehat{V}_T[k - N_A + 2] & \dots & V_T[k - 1] & V_T[k] \\ \widehat{\alpha}[k - N_A + 1] & \widehat{\alpha}[k - N_A + 2] & \dots & \widehat{\alpha}[k - 1] & \alpha[k] \\ \widehat{\beta}[k - N_A + 1] & \widehat{\beta}[k - N_A + 2] & \dots & \widehat{\beta}[k - 1] & \beta[k] \\ \widehat{Q}[k - N_A + 1] & \widehat{Q}[k - N_A + 2] & \dots & \widehat{Q}[k - 1] & Q[k] \\ \widehat{\dot{\alpha}}[k - N_A + 1] & \widehat{\dot{\alpha}}[k - N_A + 2] & \dots & \widehat{\dot{\alpha}}[k - 1] & \dot{\alpha}[k] \\ \widehat{\delta}_T[k - N_A + 1] & \widehat{\delta}_T[k - N_A + 2] & \dots & \widehat{\delta}_T[k - 1] & \delta_T[k] \\ \widehat{\delta}_E[k - N_A + 1] & \widehat{\delta}_E[k - N_A + 2] & \dots & \widehat{\delta}_E[k - 1] & \delta_E[k] \end{bmatrix}, \quad (19)$$

$$\widehat{\mathbf{Y}}^T[k] = [\widehat{a}_Z[k - N_A + 1] \quad \widehat{a}_Z[k - N_A + 2] \quad \dots \quad \widehat{a}_Z[k - 1] \quad \widehat{a}_Z[k]].$$

The training is done by minimizing the accumulated quadratic error between $\widehat{a}_Z = \bar{a}_Z + \widehat{a}_{Zb}$ and the estimated ANN vertical accelerations \underline{a}_Z , that is, minimizing the cost function $W(\boldsymbol{\eta}) = \boldsymbol{\varepsilon}^T(\boldsymbol{\eta}) \cdot \boldsymbol{\varepsilon}(\boldsymbol{\eta}) \in \mathbf{R}$, where

$$\boldsymbol{\varepsilon}(\boldsymbol{\eta}) = \begin{bmatrix} \varepsilon[k - N_A + 1, \boldsymbol{\eta}] \\ \vdots \\ \varepsilon[k, \boldsymbol{\eta}] \end{bmatrix} \quad (20)$$

$$\times \begin{bmatrix} \widehat{a}_z[k - N_A + 1] - \underline{a}_z[k - N_A + 1, \boldsymbol{\eta}] \\ \vdots \\ \widehat{a}_z[k] - \underline{a}_z[k, \boldsymbol{\eta}] \end{bmatrix},$$

where the line subscript is used to identify the ANN estimations, and where $\boldsymbol{\eta} \in R^m$ is the parameter vector

$$\boldsymbol{\eta}^T = [\eta_1, \dots, \eta_m]$$

$$= [a_{11}, \dots, a_{p1}, \dots, a_{1N}, \dots, a_{pN}, b_1, \dots, b_N,$$

$$w_{11}, \dots, w_{1n}, \dots, w_{N1}, \dots, w_{Nm}, c_1, \dots, c_p]$$

is as an approximation of the function $f_j(\mathbf{X})$; that is, $|F_j(\mathbf{X}) - f_j(\mathbf{X})| < \delta$, where $\varphi(\cdot)$ is a bounded monotonically increasing activation function. Figure 7 shows the general structure of the ANN and its interconnections.

To develop an iterative adaptive online algorithm, one learning epoch at each sample time is performed. This is done using a sliding window of data. This moving window has a constant length in time, but as new data is measured, the oldest data is discarded and the new is added as the most recent (concept similar to shift registering). Using the concept of sliding mode learning, the ANN is kept permanently in the training stage adapting its parameters to reflect changes in the dynamics of the aircraft.

For vertical acceleration estimation $\widehat{a}_Z = \bar{a}_Z + \widehat{a}_{Zb}$ (reducing the ANN to one single output), the sliding windows of input and target data, $\widehat{\mathbf{X}}[k] \in R^{N_A \times n}$ and $\widehat{\mathbf{Y}}[k] \in R^{N_A}$, are defined as follows, where N_A sets the window's length:

with $m = p \times n + N \times n + N + p$. Using a Newton-type search method, the following iterative update is defined as

$$\boldsymbol{\eta}[k + 1] = \boldsymbol{\eta}[k] - [\nabla^2 W(\boldsymbol{\eta}[k])]^{-1} \cdot \nabla^T W(\boldsymbol{\eta}[k]), \quad (22)$$

where $\nabla W(\boldsymbol{\eta}[k])$ and $\nabla^2 W(\boldsymbol{\eta}[k])$ are the gradient and the Hessian of the cost function W with respect to the parameter vector, defined as follows:

$$\nabla W(\boldsymbol{\eta}) = \begin{bmatrix} \frac{\partial W}{\partial \eta_1} & \dots & \frac{\partial W}{\partial \eta_m} \end{bmatrix},$$

$$\nabla^2 W(\boldsymbol{\eta}) = \begin{bmatrix} \frac{\partial^2 W}{\partial \eta_1^2} & \dots & \frac{\partial^2 W}{\partial \eta_1 \partial \eta_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 W}{\partial \eta_m \partial \eta_1} & \dots & \frac{\partial^2 W}{\partial \eta_m^2} \end{bmatrix}. \quad (23)$$

These quantities are approximated numerically at each iteration where the error vector $\boldsymbol{\varepsilon}(\boldsymbol{\eta})$ can be approximated by a first-order Taylor series

$$\boldsymbol{\varepsilon}(\boldsymbol{\eta} + \Delta \boldsymbol{\eta}) \approx \boldsymbol{\varepsilon}(\boldsymbol{\eta}) + J(\boldsymbol{\eta}) \cdot \Delta \boldsymbol{\eta}, \quad (24)$$

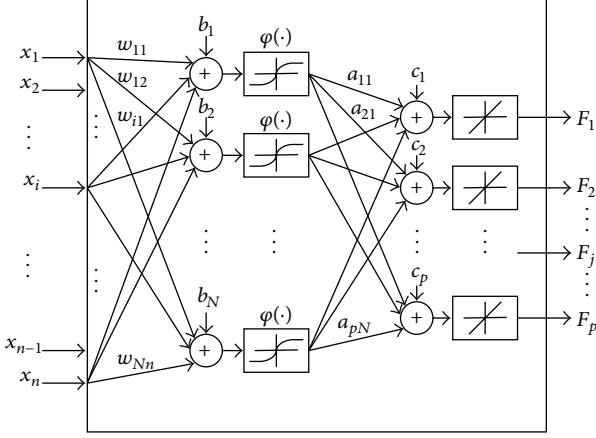


FIGURE 7: Structure of ANN.

where the rows of $\mathbf{J}(\boldsymbol{\eta})$, the Jacobian of $\boldsymbol{\varepsilon}(\boldsymbol{\eta})$, include the gradients of each element of $\boldsymbol{\varepsilon}(\boldsymbol{\eta})$ with respect to the parameter vector $\boldsymbol{\eta}$ as follows:

$$\mathbf{J}(\boldsymbol{\eta}) = \begin{bmatrix} \nabla \varepsilon [k - N_A + 1, \boldsymbol{\eta}] \\ \vdots \\ \nabla \varepsilon [k, \boldsymbol{\eta}] \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial \varepsilon [k - N_A + 1, \boldsymbol{\eta}]}{\partial \eta_1} & \dots & \frac{\partial \varepsilon [k - N_A + 1, \boldsymbol{\eta}]}{\partial \eta_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial \varepsilon [k, \boldsymbol{\eta}]}{\partial \eta_1} & \dots & \frac{\partial \varepsilon [k, \boldsymbol{\eta}]}{\partial \eta_m} \end{bmatrix}. \quad (25)$$

Then, the cost function $W(\boldsymbol{\eta})$, evaluated at $\boldsymbol{\eta} + \Delta\boldsymbol{\eta}$, is expressed as

$$W(\boldsymbol{\eta} + \Delta\boldsymbol{\eta}) \approx \frac{W(\boldsymbol{\eta})}{\boldsymbol{\varepsilon}^T(\boldsymbol{\eta}) \cdot \boldsymbol{\varepsilon}(\boldsymbol{\eta})} + \frac{\nabla W(\boldsymbol{\eta})}{2\boldsymbol{\varepsilon}^T(\boldsymbol{\eta}) \cdot \mathbf{J}(\boldsymbol{\eta}) \cdot \Delta\boldsymbol{\eta}} + \frac{1}{2} \Delta\boldsymbol{\eta}^T \cdot \frac{2\mathbf{J}^T(\boldsymbol{\eta}) \cdot \mathbf{J}(\boldsymbol{\eta})}{\nabla^2 W(\boldsymbol{\eta})} \cdot \Delta\boldsymbol{\eta}. \quad (26)$$

From (26), the cost function's gradient $\nabla W(\boldsymbol{\eta}[k])$ and Hessian $\nabla^2 W(\boldsymbol{\eta}[k])$ are then obtained as functions of the error vector $\boldsymbol{\varepsilon}(\boldsymbol{\eta})$ and its Jacobian $\mathbf{J}(\boldsymbol{\eta})$. Then, (22) is rewritten as

$$\boldsymbol{\eta}[k+1] = \boldsymbol{\eta}[k] - \left[2\mathbf{J}^T(\boldsymbol{\eta}[k]) \cdot \mathbf{J}(\boldsymbol{\eta}[k]) \right]^{-1} \cdot \left[2\boldsymbol{\varepsilon}^T(\boldsymbol{\eta}[k]) \cdot \mathbf{J}(\boldsymbol{\eta}[k]) \right]^T. \quad (27)$$

As a way to improve the numerical search, the Levenberg approach is chosen (see [40]), where a factor called Levenberg adaptive parameter λ is introduced. This method permits a transition from a steepest descent to a quadratic method

TABLE 3: Closed-loop infinity norm associated to each uncertainty.

Parameter	$\ \mathbf{F}_1(\hat{\mathbf{P}}^1, \bar{\mathbf{K}}^1)\ _\infty$
C_{mq}	0.3846
$C_{m\delta e}$	0.2451
$C_{l\beta}$	0.0100
C_{nr}	0.3816
$C_{n\delta r}$	0.1674
All	0.5617

when closer to the minimum. Based on this approach, (27) is modified as

$$\boldsymbol{\eta}[k+1] = \boldsymbol{\eta}[k] - \left[2\mathbf{J}^T(\boldsymbol{\eta}[k]) \cdot \mathbf{J}(\boldsymbol{\eta}[k]) + \lambda \mathbf{I} \right]^{-1} \cdot \left[2\boldsymbol{\varepsilon}^T(\boldsymbol{\eta}[k]) \cdot \mathbf{J}(\boldsymbol{\eta}[k]) \right]^T. \quad (28)$$

Under Levenberg's logic, this adaptive parameter is adjusted during each iteration based on the cost function evolution. In this paper, the logic is modified to reset the Levenberg's adaptive parameter, that is, $\lambda[k]$, when reaching predefined extreme values to avoid unbounded growing specially when the cost function has reached a minimum and consecutive iterations remain closer to it (a common occurrence in one-shot offline minimizations). This modification is required to keep the ANN under permanent training. The modification is as follows:

Start with $\lambda[k] = \lambda_0$

$$\text{IF } W([k+1]) < W([k]) \longrightarrow \lambda[k+1] = \frac{\lambda[k]}{\lambda_{\text{dec}}}$$

$$\text{IF } \lambda[k+1] < \lambda_{\text{min}} \longrightarrow \lambda[k+1] = \lambda_0$$

$$\text{IF } W([k+1]) \geq W([k]) \longrightarrow \lambda[k+1] = \lambda[k] \cdot \lambda_{\text{inc}}$$

$$\text{IF } \lambda[k+1] > \lambda_{\text{max}} \longrightarrow \lambda[k+1] = \lambda_0, \quad (29)$$

where $0 < \lambda_{\text{min}} < \lambda_{\text{max}}$ are the minimum and maximum allowed values, and $\lambda_{\text{dec}} > 0$ and $\lambda_{\text{inc}} > 0$ are the decrement and increment correction factors. All these factors are empirically obtained after extensive simulations.

6. Simulation and Testing

This section is devoted to test previous designs in simulation, by employing stored flight test data, and also by comparing the designed FCS performance in contrast with a COTS autopilot under similar conditions, for the same aircraft. The manufacture of COTS system cannot be revealed due to contractual obligations.

6.1. ANN Simulation Results. The ANN estimation capability is evaluated for a window length of $N_A = 400$ (samples), which corresponds to 10 seconds of past flight for a sampling

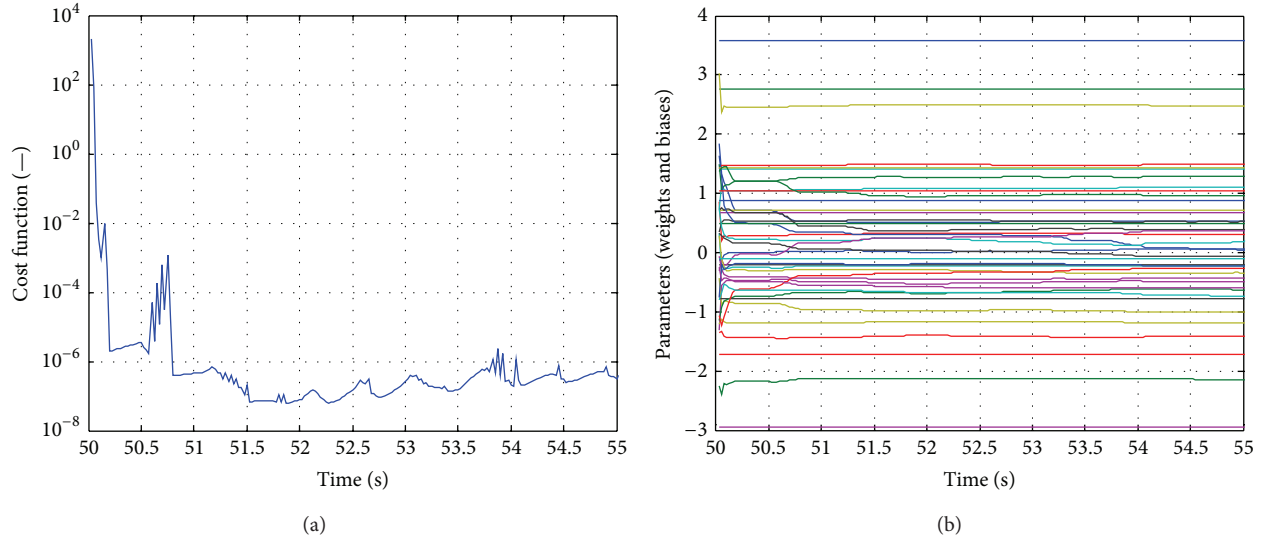


FIGURE 8: Cost function minimization, no noise (a). Parameter evolution, no noise (b).

interval $dt = 1/40$ seconds. For the logic defined in (29), $\lambda_0 = 0.001$, $\lambda_{\max} = 1$, $\lambda_{\min} = 0.00001$, $\lambda_{\text{dec}} = 3$, and $\lambda_{\text{inc}} = 2$.

Two simulations are run for a 33% Yak-54 UAS, with and without measurement noise. Figures 8 and 9 show the convergence of the parameters, the cost function minimization, the Levenberg adaptive parameter, and the measured and estimated vertical acceleration, considering noise. Figures 10 and 11 show the same information for the case without noise.

By using the Delta method, which is a numerical technique to approximate partial derivatives, stability and control derivatives can be extracted.

The standard application of this technique, in previous cited articles, is generally offline postprocessing analysis. The idea presented here is to allocate a specific logic and to compute these approximate partial derivatives online. Two identical structures of the ANN are kept in parallel. They are fed incoming measurements, similar to the ANN under training. Also, these two structures are updated with the current weight and biases from the ANN. One of these structure is fed with a perturbed input (i.e., measured angle of attack plus a small added perturbation), while the second structure is kept fed by unperturbed inputs. By taking the quotient between the differences of the two structures outputs, with respect to the magnitude of the added perturbation, an approximation of the partial derivative is obtained in real time.

For this case, the partial derivative of the vertical acceleration with respect to some input of the ANN is obtained. These quantities can then be used to calculate more specific stability and control derivatives by converting from body to stability coordinates. This allows the capability of detecting dynamic changes due to internal or external changing conditions, or due to failures, by monitoring the time tendency of these values.

By comparing Figures 8 and 9, with Figures 10 and 11, the effect of measurement noise is clear. Cost function values are considerably higher. Also, the parameter vector does not

settle permanently and it adjusts continuously due to the inherent variability condition of the Levenberg parameter.

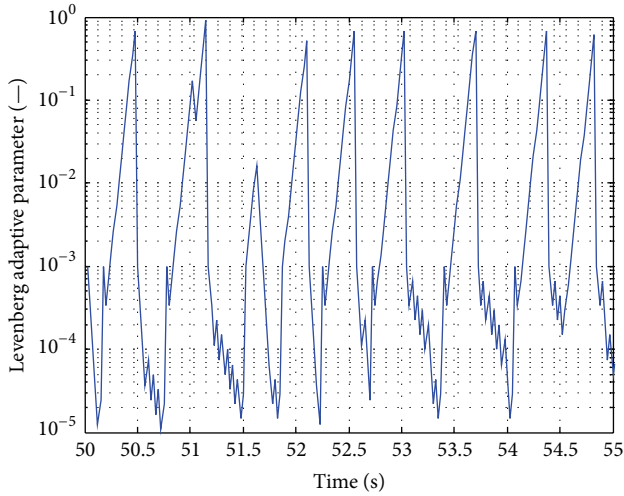
As shown in Figure 11(b), a fairly good prediction is achieved. Further precaution needs to be observed to avoid deep changes, as seen between seconds 54 and 55 when a steep readjustment occurred. Using a desktop, each iteration took less than 1.2 seconds; this means that a dedicated microprocessor must be used for real-time estimations.

6.2. Adaptive Artificial Neural Network Testing. To verify the performance of ANN's adaptive capability in the modeling of unsteady aerodynamic forces, actual 33% scale Yak-54 UAS flight test data from the 2009 flight test at Pegasus Airfield in Antarctica is used. The data was collected with a sampling frequency of 10 (Hz). For comparison and validation purposes, the same ANN for Z-axis acceleration is trained and tested with and without adaptiveness. This ANN is constructed from twenty cells ($M = 20$) in the hidden layer and fed with same inputs as defined earlier for online ANN. Figure 12 shows the flight test data and the estimations with and without adaptiveness.

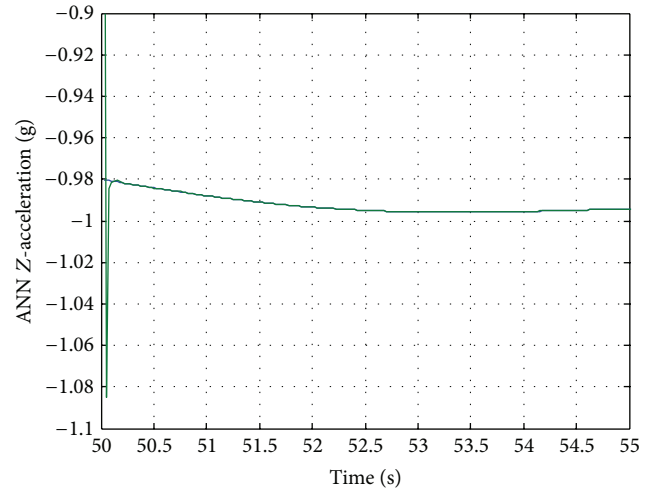
The flight had a well-defined pattern. From a prolonged circular autonomous period of flight, the aircraft is switched to remote control for landing around second 1145. During autonomous flight, the nonadaptive ANN is trained. It is then tested in both the remaining autonomous flight segment and during the remote controlled period. The adaptive ANN is tested in the same interval as the nonadaptive ANN.

Figure 12 shows a close estimation of both types of ANN before mode switching. After changing control sources, a significant dynamic behavior experienced by the aircraft left nonadaptive ANN outdated.

6.3. Gain Scheduling Simulation Results. To assess the gain scheduling techniques, hypothetical failures are induced through changes in the numerical value of stability and

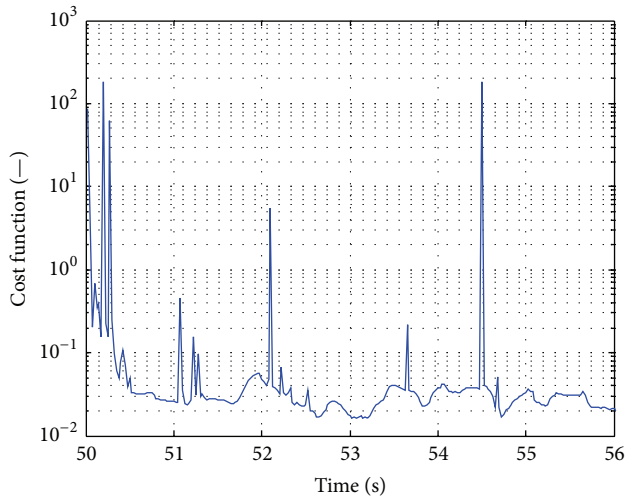


(a)

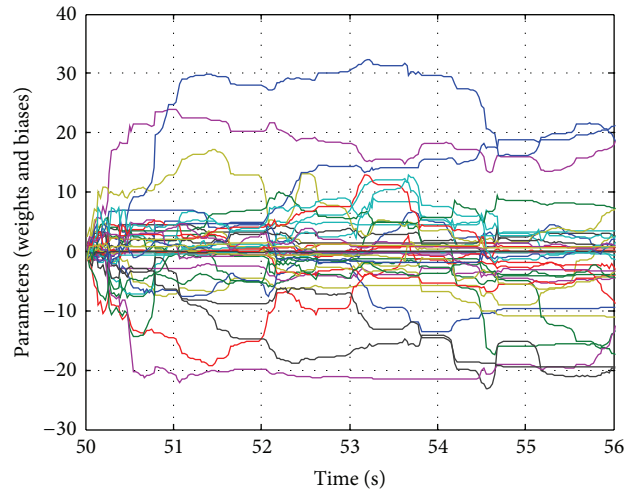


(b)

FIGURE 9: Levenberg adaptive parameter, no noise (a). Vertical acceleration, no noise (b).



(a)



(b)

FIGURE 10: Cost function minimization, with noise (a). Parameter evolution, with noise (b).

control derivatives. The gain scheduling adapts controllers to guarantee the aircraft stability and performance in the event of an adverse condition (e.g., servo failure or damage). Extreme changes could be a result of system failure or a result of damage to the aircraft. An adaptive flight control system must have the potential to adjust for failure or damage to maintain control in flight and meet the required handling qualities that allow for safe recovery of the airplane. As a condition for successful gain scheduling, the FCS is expected to achieve a similar level of performance and required uncertainty management before and after detection of dynamic changes.

To replicate a major change in the lateral-directional dynamics of UAS due to a failure or unsteady aerodynamics,

at 120 seconds of a horizontal level flight, the numerical value of $C_{n\beta}$ (Table 1) is reduced by a factor of 4. This reflects a sudden change in the UAS dynamics, affecting the stability of UAS directionally. After inducing this change, the current controller loses its stabilizing capability and oscillation can be traced in the aircraft's lateral-directional states. As it was demonstrated in Section 6.2, the adaptive ANN allows identification of sudden changes in the aircraft dynamics in real time.

At 138 seconds and after detection of an unexpected change in the aircraft's stability and control derivative (i.e., change in $C_{n\beta}$), the adaptive controller replaced the old controller with a new one and successfully mitigated sudden

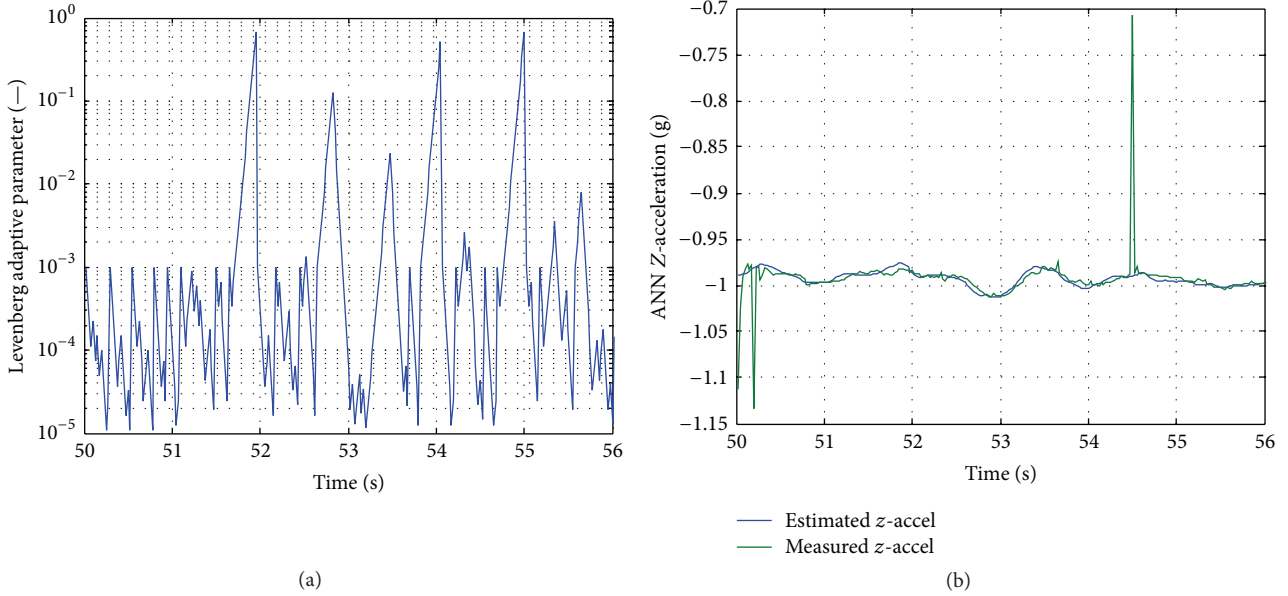


FIGURE 11: Levenberg adaptive parameter, with noise (a). Vertical acceleration, with noise (b).

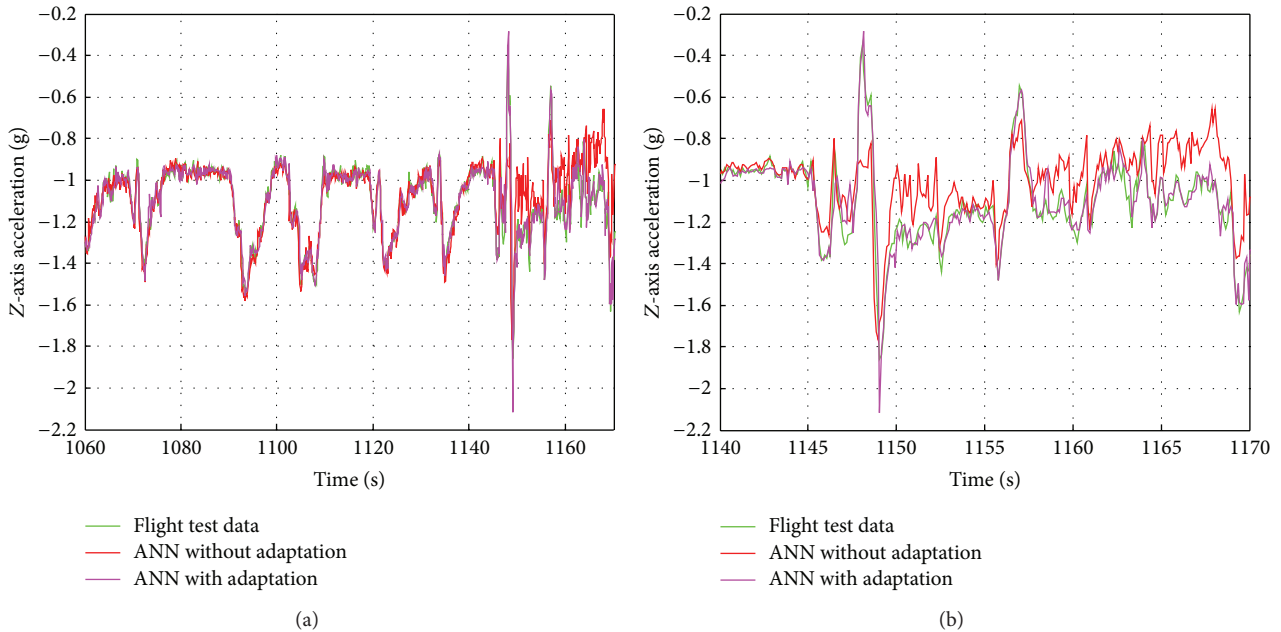


FIGURE 12: Z-axis measured and predicted acceleration: larger flight portion (a); zoom after switching control (b).

change in the aircraft’s dynamics. The growing amplitude of oscillation is damped out and the roll and yaw rates and sideslip angle are effectively controlled.

The resultant period of time taken to implement the gain scheduling (18 seconds) is affected by the fact that the failure does not show up immediately and the vehicle remains mostly unaffected. The induced failure is an alteration of the rate of change in yaw rate with respect to changes in sideslip angle, that is, $C_{n\beta}$, so for an original constant and close to zero sideslip (horizontal level flight), no immediate effect is

observed. After sideslip angle starts to change, the failure has a multiplicative effect and grows faster. This is evident after second 130.

It is important to point out that although the closed loop remains (linearly) stable, the uncertainty robustness is lost, as the infinity norm grows over the unity $\|F_i(\hat{P}^1, \bar{K}^1)\|_\infty = 4.6355$. In other words, a well-designed flight control system is robust to a fairly large range of uncertainty or changes in aircraft dynamics, however, if the changes become more extreme the performance degrades and the aircraft may

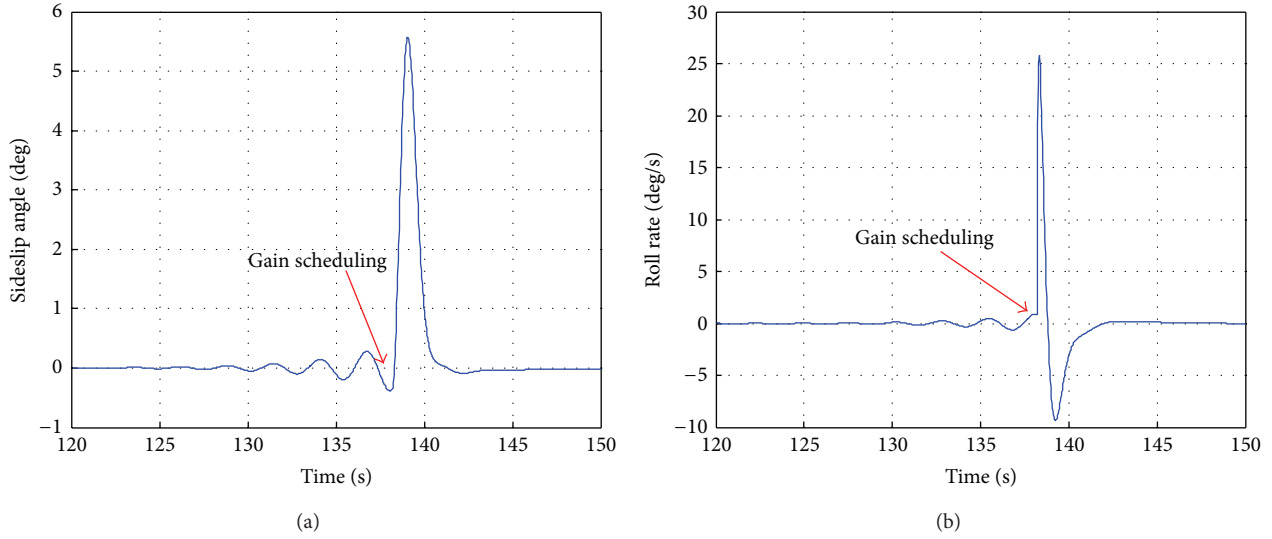


FIGURE 13: Sideslip angle (a). Roll rate (b).

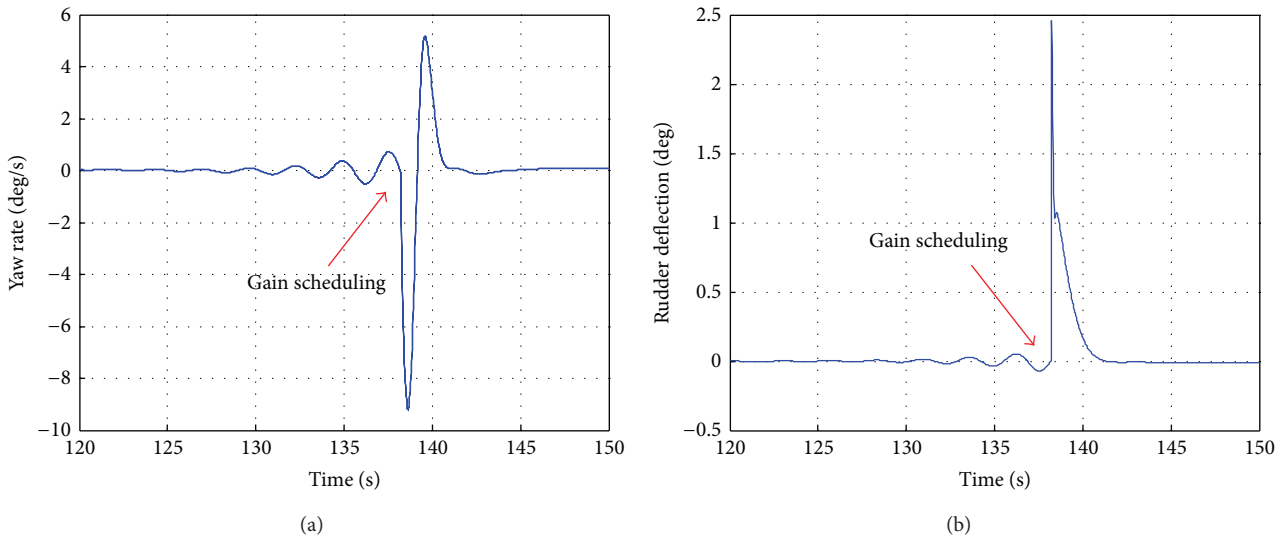


FIGURE 14: Yaw rate (a). Rudder deflection (b).

go unstable. An adaptive system has the ability to readjust the controller to achieve desired performance or regain robustness about the new point.

In transition between controllers and to avoid significant sensitivity, the gain scheduling must be coupled with the ANN output. Figures 13 and 14 show the failure and consequent gain scheduling effects, on primarily involved states. All simulations were performed with the full FCS.

To broaden the applicability of FCS technologies and to extend assessment process, a failure which degrades aircraft's longitudinal stability is modeled. It is assumed that at 200 seconds of a horizontal level flight, a failure in the aircraft caused a major change in the magnitude of $C_{m\alpha}$ (Table 1). The numerical value of $C_{m\alpha}$ is reduced by a factor of 10. As it is shown in Figures 15, 16, and 17 and after the failure took place, the longitudinal robust stability is severely affected; that is, $\|F_l(\hat{P}^1, \bar{K}^1)\|_\infty = 0.8895$.

Detection of the failure, calculation of a new controller, and gain scheduling are assumed to happen after 5 seconds (200–205 (sec)). At 205 seconds, the new controller is online modifying control surface commands. Severe impact is seen in all major longitudinal variables, as shown in Figures 15, 16, and 17.

The abrupt change in the angle of attack stability coefficient ($C_{m\alpha}$) increases the pitch angle and subsequently reduces the airspeed. To mitigate these effects, the elevator is commanded to quickly recover the equilibrium about pitch axes and restoring the pitch angle. To restore the airspeed, the throttle is also commanded. As demonstrated with the simulated failure, the system without adaptation cannot achieve the desired tracking or performance.

6.4. Functionality Tests and Comparison with Flight Test Data. The purpose of this subsection is to assess different

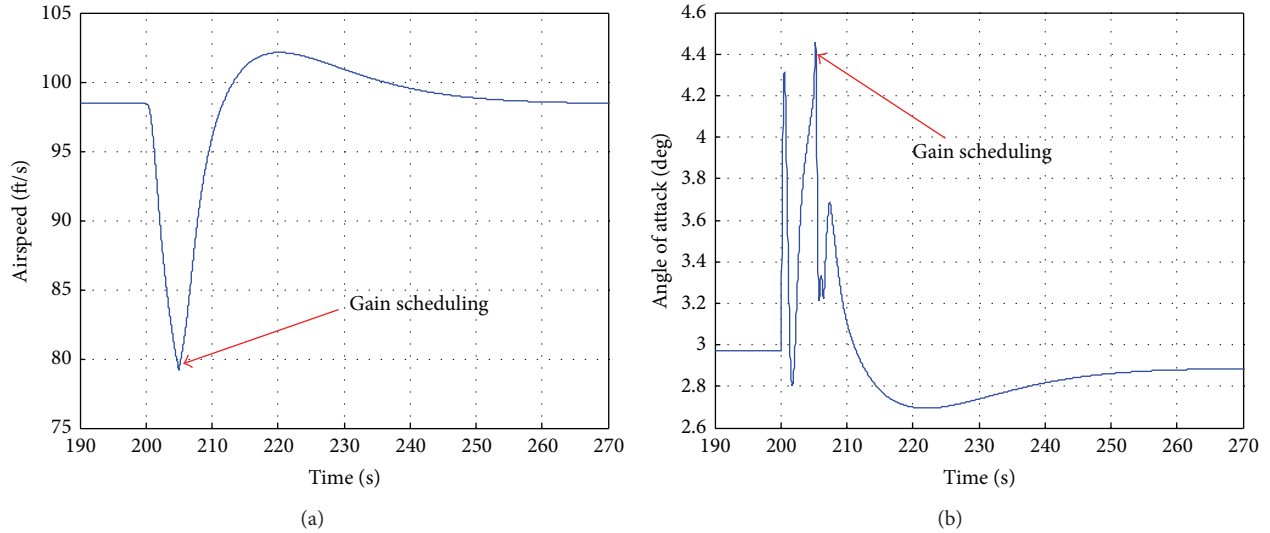


FIGURE 15: Airspeed (a). Angle of attack (b).

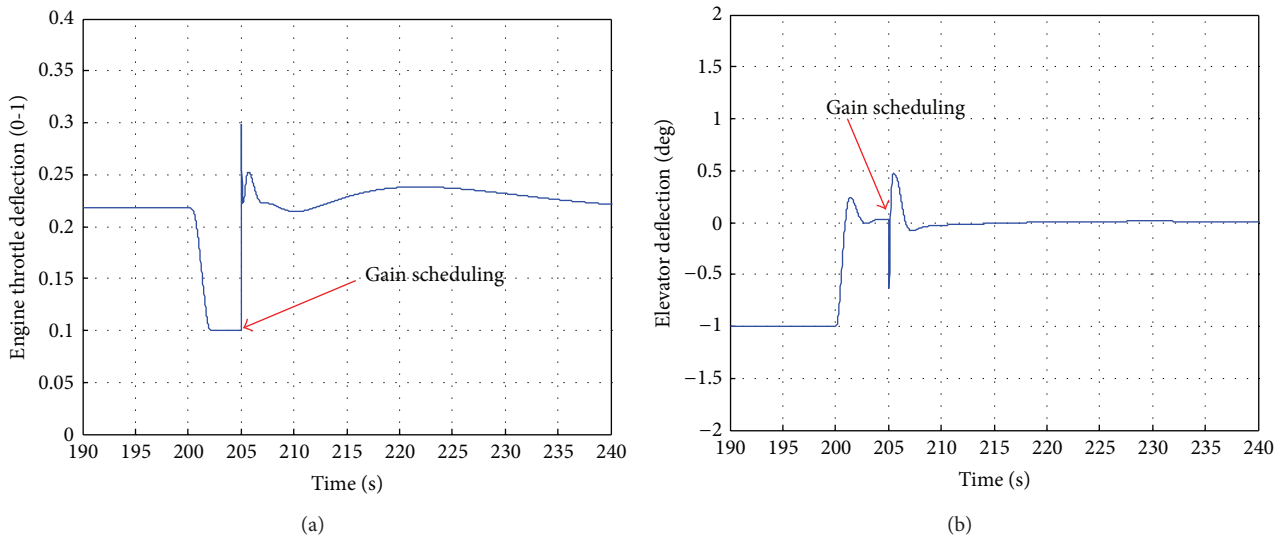


FIGURE 16: Engine throttle (a). Elevator deflection (b).

functionalities of the designed FCS and to compare its performance in closed loop with autonomous flight test data obtained using a COTS autopilot system. It is known that the COTS system uses an H-infinity controller. For comparison purposes, similar flight conditions are induced in the simulation, and same waypoints/trajectory are used for path following. Superficially, performances of the designed FCS and the COTS autopilot are compared in the following categories:

- (i) extended Kalman filter performance (flight measurements versus EKF estimation);
- (ii) inner loop: H-infinity robust controller performance (e.g., the maximum commanded bank angle);
- (iii) outer loop: accuracy of guidance logic and path following in tight turns in presence of crosswind (e.g., deviation from the desired trajectory);

(iv) airspeed and altitude control;

(v) saturation of controls.

6.4.1. Extended Kalman Filter Testing. A slow update rate GPS position sampled at 2 (Hz) is employed to test the smooth interpolation capability of the EKF. Figure 18 shows the measured and estimated geographical position and altitude.

6.4.2. Flight Control System Testing. As part of the comparison, inner and outer loops and extended Kalman filter, as defined before, are included in testing. In this subsection, no gain scheduling and no ANN estimations are performed. These two features are tested independently and presented in previous two subsections.

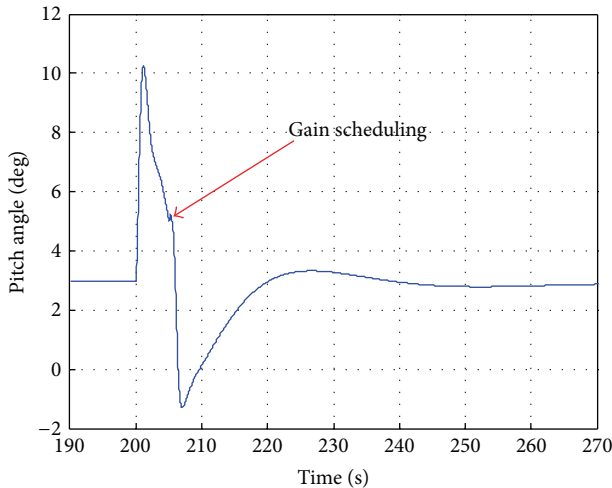


FIGURE 17: Pitch angle.

The FCS controls the aircraft to follow a trajectory autonomously flown by a 33% Yak-54 UAS equipped with a COTS autopilot. This system contains an H-infinity robust controller in the inner loop, an outer loop logic for waypoint following, and an extended Kalman filter, running at 10 (Hz). Its sensors include a GPS, an IMU, a 3D accelerometer, a 3D gyro, a 3D magnetometer, a barometer, and a pitot tube.

For the similarity of the flight test and simulation comparisons, same initial conditions are extracted and used from the flight test data. To replicate external disturbance, COTS estimated wind information was also injected.

Figure 19 shows 3D views, and Figures 20 and 21 show the control signals for both the designed FCS and COTS autopilot. The top view of 3D simulation in Figure 19 demonstrates superior performance of new nonlinear guidance logic in guiding UAS on curved and tight trajectories in presence of a crosswind. The wind was blowing from the west, and the maximum deviation from the desired trajectory happened when the UAS equipped with a COTS autopilot was in a crosswind (magenta line Figure 19).

By design, the COTS autopilot does not command rudder. This imposed extra demand on the aileron causing higher banking angles when initiating turns. Despite dissimilarity of COTS system and the designed FCS, some basic but important comparisons can be made. As it is shown in Figures 20 and 21, the COTS autopilot and the designed FCS are not subject of control saturation issues.

Figure 22(a) shows the comparison between the COTS autopilot and the designed FCS in airspeed command following. For the sake of unbiased comparison, the COTS autopilot wind estimation is used in the simulation 14.1 (ft/sec) from 291°. The airspeed command has a step increment around 380 seconds.

The ground speed is shown in Figure 22(b) in a wider portion of the flight. The error in airspeed control of the COTS autopilot is seen as the result of bias. This presumption is confirmed by adding a bias of about 10–12 (ft/sec) to the measured airspeed and disabling the FCS's Kalman filter for

an even comparison. After inducing the bias, ground speeds achieve higher similarity.

Figure 23 shows the attitude angles. It is important to notice that the extended Kalman filter takes a few seconds to converge from the given initial conditions affecting the first part of the estimation, but showing a good command following after possible transients.

7. Conclusions

An adaptive control comprising a robust inner controller, stable outer nonlinear guidance logic, and neural network fault detection is designed to provide increased resiliency to mitigate effects of on-board adverse conditions and unsteady aerodynamics. The aerodynamic models of UAS associated with hazardous weather or on-board adverse conditions are inherently nonlinear and unsteady. Under these conditions, although the closed-loop H-infinity is still linearly stable, the controller does not have uncertainty robustness and the infinity norm can grow over unity. A modified artificial neural network is successfully implemented for real-time nonlinear estimations of the aerodynamic forces. The Levenberg's adaptive parameter resets when reaching predefined extreme values. This modification allows avoiding unbounded growing when the cost function has reached a minimum. The sliding mode learning concept is used to keep the artificial neural network in learning phase continuously. A dynamic learning process of artificial neural networks increases the robustness against on-board adverse conditions and external disturbances. New nonlinear guidance logic which is extended from the lateral plane to the longitudinal plane has demonstrated superior performance in guiding unmanned air systems on curved trajectories in presence of external disturbances such as crosswind. The new logic adds new features ensuring its stability for any location and any attitude of the aircraft. The flight control system shows a satisfactory level of adaptability and resilience which is achieved by the adaptive nonlinear modeling, noise rejection, and biases attenuation. The robustness is assured by the proper gain scheduling between predefined trim points and for identified dynamic changes. It was found that in transition between gains and to avoid significant sensitivity, the gain scheduling must be coupled with the ANN output.

Appendices

A. Aircraft Dynamic Modeling

Based on a Newton-Euler modeling approach, the aircraft dynamic model comprises a 6-DOF rigid-body set of equations for a fixed-wing aircraft in flat earth. Referred to the aircraft's center of gravity CG, as shown in Figure 1(b), it is defined in four interrelated coordinate systems, that is, the inertial coordinate system \bar{I} aligned with the inertial local flat earth coordinate system I , the body coordinate system \bar{B} fixed to the aircraft, the wind coordinate system \bar{W} oriented

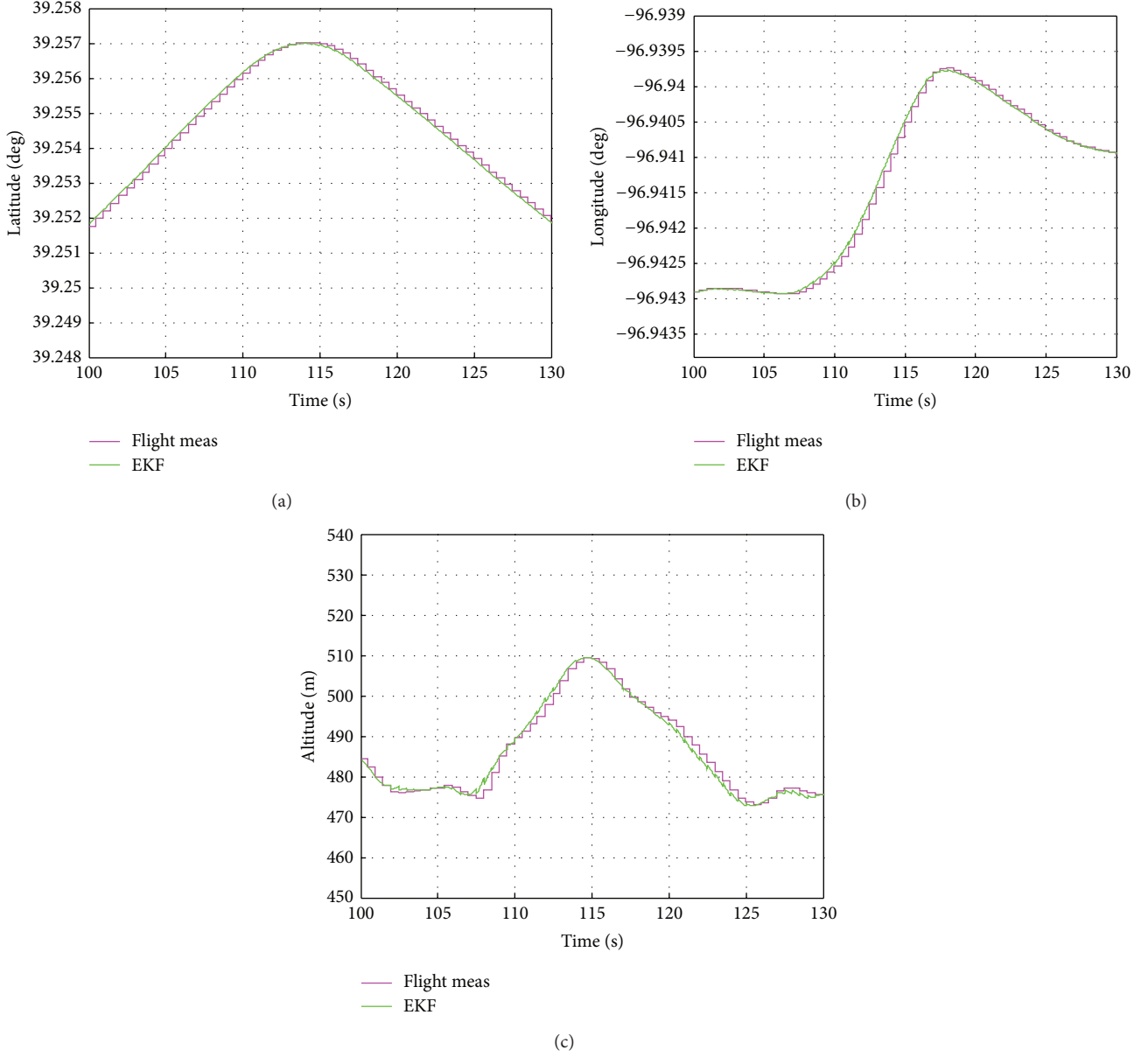


FIGURE 18: Latitude (a). Longitude (b). Altitude (c).

toward the relative wind, and the stability coordinate system \bar{S} oriented as shown. Consider

$$\dot{V}_T = \frac{(U\dot{U} + V\dot{V} + W\dot{W})}{V_T},$$

$$\dot{\alpha} = \frac{(U\dot{W} - W\dot{U})}{(UU + WW)},$$

$$\dot{\beta} = \frac{(V_T\dot{V} - V\dot{V}_T)}{(V_T^2 \cos \beta)},$$

$$\dot{\phi} = P + \tan \theta (Q \sin \phi + R \cos \phi),$$

$$\dot{\theta} = Q \cos \phi - R \sin \phi,$$

$$\dot{\psi} = \frac{(Q \sin \phi + R \cos \phi)}{\cos \theta},$$

$$\dot{P} = (J_{xz} [J_x - J_y + J_z] PQ - [J_z (J_z - J_y) + J_{xz}^2] QR + J_z L_A + J_{xz} N_A) \times (J_x J_z - J_{xz}^2)^{-1},$$

$$\dot{Q} = \frac{[J_z - J_x] PR - J_{xz} [P^2 - R^2] + M_A}{J_x J_z - J_{xz}^2},$$

$$\dot{R} = ([(J_x - J_y) J_x + J_{xz}^2] PQ - J_{xz} [J_x - J_y + J_z] QR + J_{xz} L_A + J_x N_A) \times (J_x J_z - J_{xz}^2)^{-1},$$

$$\begin{aligned}\dot{\delta}_t &= \frac{\delta_t}{\tau_t} + \frac{\delta_t^{\text{cmd}}}{\tau_t}, \\ \dot{\delta}_e &= \frac{\delta_e}{\tau_e} + \frac{\delta_e^{\text{cmd}}}{\tau_e}, \\ \dot{\delta}_a &= \frac{\delta_a}{\tau_a} + \frac{\delta_a^{\text{cmd}}}{\tau_a}, \\ \dot{\delta}_r &= \frac{\delta_r}{\tau_r} + \frac{\delta_r^{\text{cmd}}}{\tau_r},\end{aligned}$$

$$\begin{aligned}\dot{p}_N &= U \cos \theta \cos \psi + V (-\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi) \\ &\quad + W (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) - w_N, \\ \dot{p}_E &= U \cos \theta \sin \psi + V (\cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi) \\ &\quad + W (-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) - w_E, \\ \dot{p}_H &= U \sin \theta - V \sin \phi \cos \theta - W \cos \phi \cos \theta - w_H,\end{aligned}\tag{A.1}$$

where moments of inertia J_x, J_y, J_z , product of inertia J_{xz} , mass m , and gravity acceleration g are assumed constant. Body velocities U, V, W in body frame \mathbf{B} are define by $U = V_T \cos \alpha \cos \beta$, $V = V_T \sin \beta$, and $W = V_T \sin \alpha \cos \beta$, and their rate of change as

$$\begin{aligned}\dot{U} &= RV - QW - g \sin \theta + \frac{(X_T + X)}{m}, \\ \dot{V} &= -RU + PW + g \sin \phi \cos \theta + \frac{Y}{m}, \\ \dot{W} &= QU - PV + g \cos \phi \cos \theta + \frac{Z}{m},\end{aligned}\tag{A.2}$$

where aerodynamic forces X^A, Y^A, Z^A and moments L^A, M^A, N^A , together with the engine thrust X_T , act in the system as controllable inputs.

B. Polynomial Time-Invariant Engine Thrust Modeling

Highly dependent on the type and configuration of the engine, thrust X^P is modeled parametrically as a polynomial function on δ_T as

$$X^P(\mathbf{x}, t) = f_{X^P}(\delta_T, t) = T_X^2 \cdot \delta_T^2(t) + T_X^1 \cdot \delta_T(t) + T_X^0,\tag{B.1}$$

where T_X^2, T_X^1, T_X^0 are constant parameters to be estimated from flight test data for the trim point \mathbf{x}_{trim} .

C. Linear and Time-Invariant Aerodynamic Coefficient Modeling

The following relations, defined in the stability axis $\bar{\mathbf{S}}$, are used to model the aerodynamic coefficients $C_D, C_L, C_m, C_y, C_l, C_n$ as

$$\begin{aligned}C_D &= C_{D0} + C_{D\alpha}\alpha + C_{D\delta e}\delta e + C_{D\dot{\alpha}}\frac{\dot{\alpha}\bar{c}}{2U_{\text{trim}}} \\ &\quad + C_{Dq}\frac{q\bar{c}}{2U_{\text{trim}}} + C_{Du}\frac{u}{U_{\text{trim}}}, \\ C_L &= C_{L0} + C_{L\alpha}\alpha + C_{L\delta e}\delta e + C_{L\dot{\alpha}}\frac{\dot{\alpha}\bar{c}}{2U_{\text{trim}}} \\ &\quad + C_{Lq}\frac{q\bar{c}}{2U_{\text{trim}}} + C_{Lu}\frac{u}{U_{\text{trim}}}, \\ C_m &= C_{m0} + C_{m\alpha}\alpha + C_{m\delta e}\delta e + C_{m\dot{\alpha}}\frac{\dot{\alpha}\bar{c}}{2U_{\text{trim}}} \\ &\quad + C_{mq}\frac{q\bar{c}}{2U_{\text{trim}}} + C_{mu}\frac{u}{U_{\text{trim}}}, \\ C_y &= C_{y\beta}\beta + C_{y\dot{\beta}}\frac{\dot{\beta}b}{2U_{\text{trim}}} + C_{yp}\frac{pb}{2U_{\text{trim}}} + C_{yr}\frac{rb}{2U_{\text{trim}}} \\ &\quad + C_{y\delta a}\delta a + C_{y\delta r}\delta r, \\ C_l &= C_{l\beta}\beta + C_{l\dot{\beta}}\frac{\dot{\beta}b}{2U_{\text{trim}}} + C_{lp}\frac{pb}{2U_{\text{trim}}} + C_{lr}\frac{rb}{2U_{\text{trim}}} \\ &\quad + C_{l\delta a}\delta a + C_{l\delta r}\delta r, \\ C_n &= C_{n\beta}\beta + C_{n\dot{\beta}}\frac{\dot{\beta}b}{2U_{\text{trim}}} + C_{np}\frac{pb}{2U_{\text{trim}}} + C_{nr}\frac{rb}{2U_{\text{trim}}} \\ &\quad + C_{n\delta a}\delta a + C_{n\delta r}\delta r,\end{aligned}\tag{C.1}$$

where p, q, r are the deviations of the body angular velocities from their constant trim values $P_{\text{trim}}, Q_{\text{trim}}, R_{\text{trim}}$, and u is the deviation from the constant trim speed U_{trim} . Aerodynamic coefficients in the stability coordinate system $\bar{\mathbf{S}}$ are transformed into body axes \mathbf{B} coefficients by a rotation in the amount of angle of attack trim value α_{trim} .

D. H-Infinity Robust Controller Design Conditions

Conditions to obtain a stabilizing H-infinity controller are

(a) $(\mathbf{A}^i, \mathbf{B}_2^i)$ stabilizable and $(\mathbf{C}_2^i, \mathbf{A}^i)$ detectable;

(b) $\text{rank}(\mathbf{D}_{12}^i) = m_2$ and $\text{rank}(\mathbf{D}_{21}^i) = p_2$;

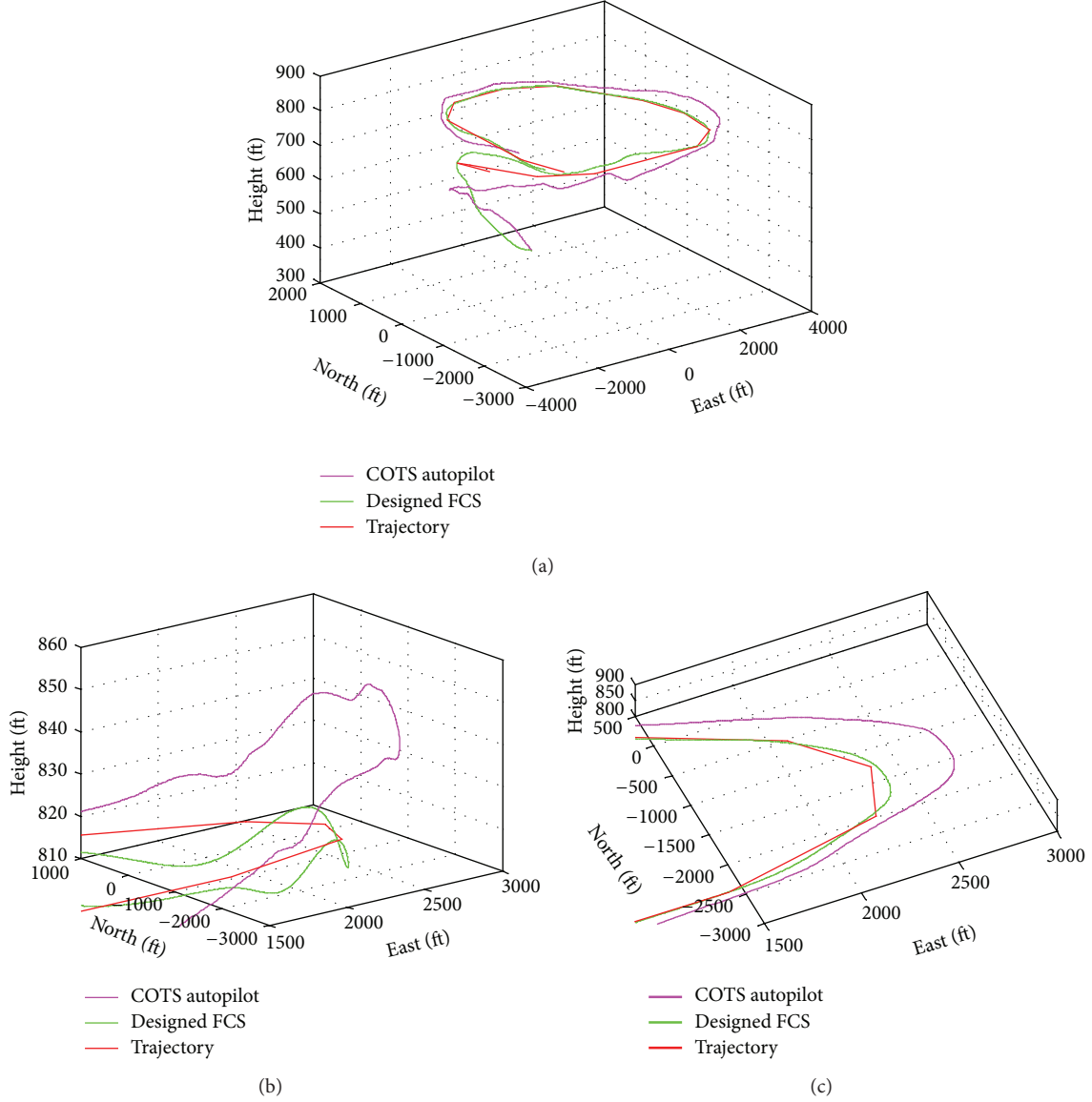


FIGURE 19: 3D full view path following (a). 3D zoom view path following (b) and (c).

(c) $\mathbf{D}_{12}^i = [0 \ I_{m_2}]^T$ and $\mathbf{D}_{21}^i = [0 \ I_{p_2}]$, and \mathbf{D}_{11}^i is expressible as

$$\mathbf{D}_{11}^i = \begin{bmatrix} \mathbf{D}_{1111}^i & \mathbf{D}_{1112}^i \\ (p_1 - m_2) \times (m_1 - p_2) & (p_1 - m_2) \times (p_2) \\ \mathbf{D}_{1121}^i & \mathbf{D}_{1122}^i \\ (m_2) \times (m_1 - p_2) & (m_2) \times (p_2) \end{bmatrix}; \quad (\text{D.1})$$

(d) $\mathbf{D}_{22}^i = 0$;

(e) $\text{rank} \begin{bmatrix} \mathbf{A}_1^{-j\omega I} & \mathbf{B}_2^i \\ \mathbf{C}_1^i & \mathbf{D}_{12}^i \end{bmatrix} = n + m_2$;

(f) $\text{rank} \begin{bmatrix} \mathbf{A}_1^{-j\omega I} & \mathbf{B}_1^i \\ \mathbf{C}_2^i & \mathbf{D}_{21}^i \end{bmatrix} = n + p_2$,

where assumption c, if not achieved by the chosen weighting matrices, can be obtained via scaling of \mathbf{u}_2 and \mathbf{y}_2 together with a unitary transformation of \mathbf{u}_1 and \mathbf{y}_1 . After previous

assumptions are met, there is $\gamma^i > 0$, for which the following relations holds:

$$\max(\bar{\sigma}[\mathbf{D}_{1111}^i \ \mathbf{D}_{1112}^i], \bar{\sigma}[\mathbf{D}_{1111}^{i T} \ \mathbf{D}_{1112}^{i T}]) < \gamma^i, \quad \mathbf{X}^i \geq 0, \mathbf{Y}^i \geq 0, \quad (\text{D.2})$$

$$\rho(\mathbf{X}^i \cdot \mathbf{Y}^i) < (\gamma^i)^2.$$

This is equivalent to the existence of an internal stabilizing controller $\bar{\mathbf{K}}^i$

$$\bar{\mathbf{K}}^i = \begin{bmatrix} \bar{\mathbf{A}}^i & \bar{\mathbf{B}}_1^i & \bar{\mathbf{B}}_2^i \\ \bar{\mathbf{C}}_1^i & \bar{\mathbf{D}}_{11}^i & \bar{\mathbf{D}}_{12}^i \\ \bar{\mathbf{C}}_2^i & \bar{\mathbf{D}}_{21}^i & 0 \end{bmatrix} \quad (\text{D.3})$$

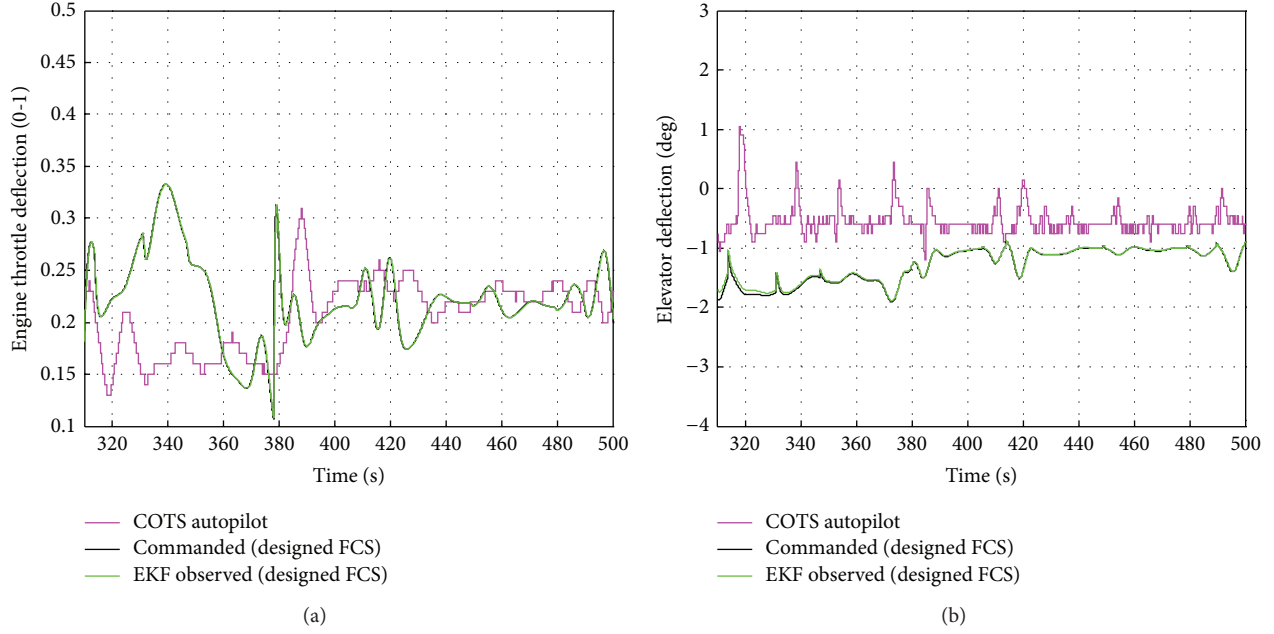


FIGURE 20: Engine throttle deflection (a). Elevator deflection (b).

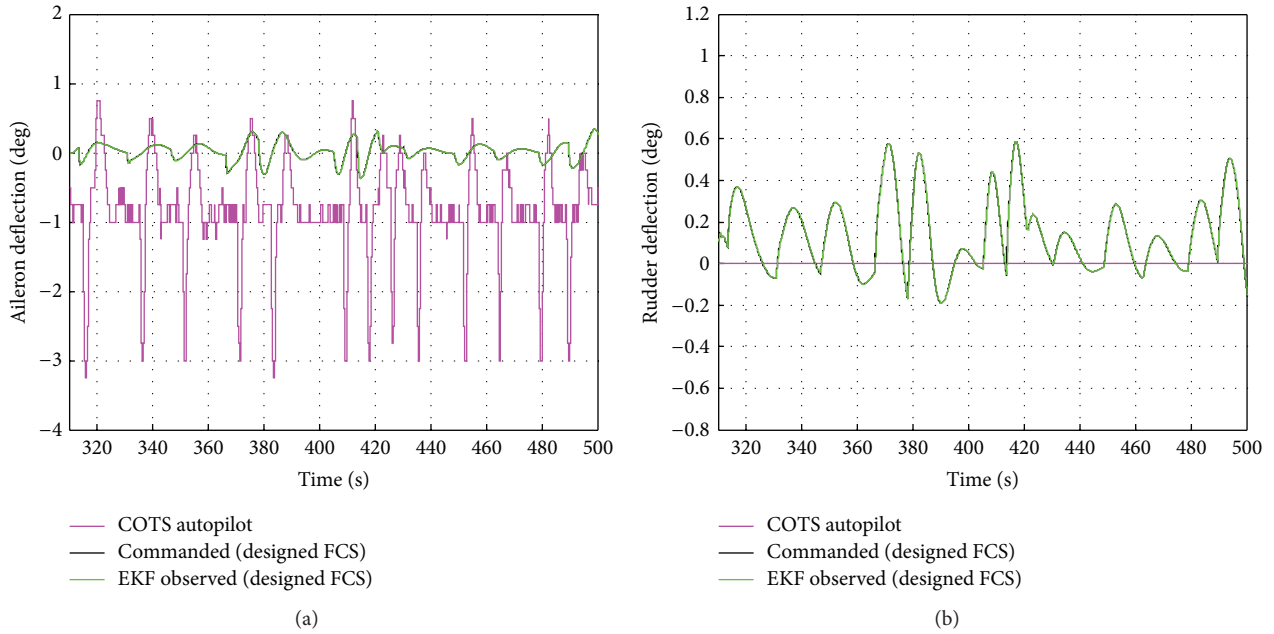


FIGURE 21: Aileron deflection (a). Rudder deflection (b).

whose components are given by

$$\bar{\mathbf{A}}^i = \mathbf{A}^i + \mathbf{H}^i \mathbf{C}^i + \bar{\mathbf{B}}_2^i (\bar{\mathbf{D}}_{12}^i)^{-1} \bar{\mathbf{C}}_1^i,$$

$$\bar{\mathbf{B}}_1^i = -\mathbf{H}_2^i + \bar{\mathbf{B}}_2^i (\bar{\mathbf{D}}_{12}^i)^{-1} \bar{\mathbf{D}}_{11}^i,$$

$$\bar{\mathbf{B}}_2^i = (\mathbf{B}_2^i + \mathbf{H}_{12}^i) \bar{\mathbf{D}}_{12}^i,$$

$$\bar{\mathbf{C}}_1^i = \mathbf{F}_2^i \mathbf{Z}^i + \bar{\mathbf{D}}_{11}^i (\bar{\mathbf{D}}_{21}^i)^{-1} \bar{\mathbf{C}}_2^i,$$

$$\bar{\mathbf{C}}_2^i = -\bar{\mathbf{D}}_{21}^i (\mathbf{C}_2^i + \mathbf{F}_{12}^i) \mathbf{Z}^i,$$

$$\bar{\mathbf{D}}_{11}^i = -\mathbf{D}_{1121}^i \mathbf{D}_{1111}^{i T} ((\gamma^i)^2 \mathbf{I} - \mathbf{D}_{1111}^i \mathbf{D}_{1111}^{i T})^{-1} \mathbf{D}_{1112}^i - \mathbf{D}_{1122}^i, \quad (\text{D.4})$$

and $\bar{\mathbf{D}}_{12}^i$ and $\bar{\mathbf{D}}_{21}^i$ are any matrices satisfying the relation $\bar{\mathbf{D}}_{12}^i \bar{\mathbf{D}}_{12}^{i T} = \mathbf{I} - \bar{\mathbf{D}}_{1121}^i ((\gamma^i)^2 \mathbf{I} - \bar{\mathbf{D}}_{1111}^i \bar{\mathbf{D}}_{1111}^{i T})^{-1} \bar{\mathbf{D}}_{1121}^{i T}$ and the

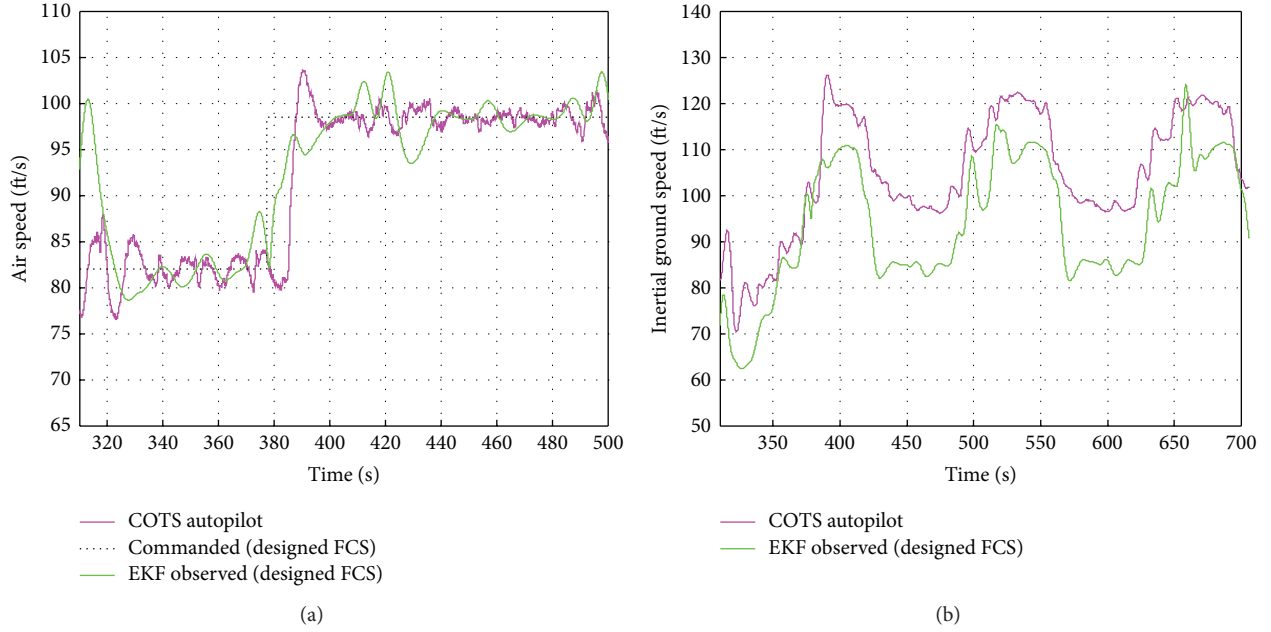


FIGURE 22: Airspeed (a). Ground speed (b).

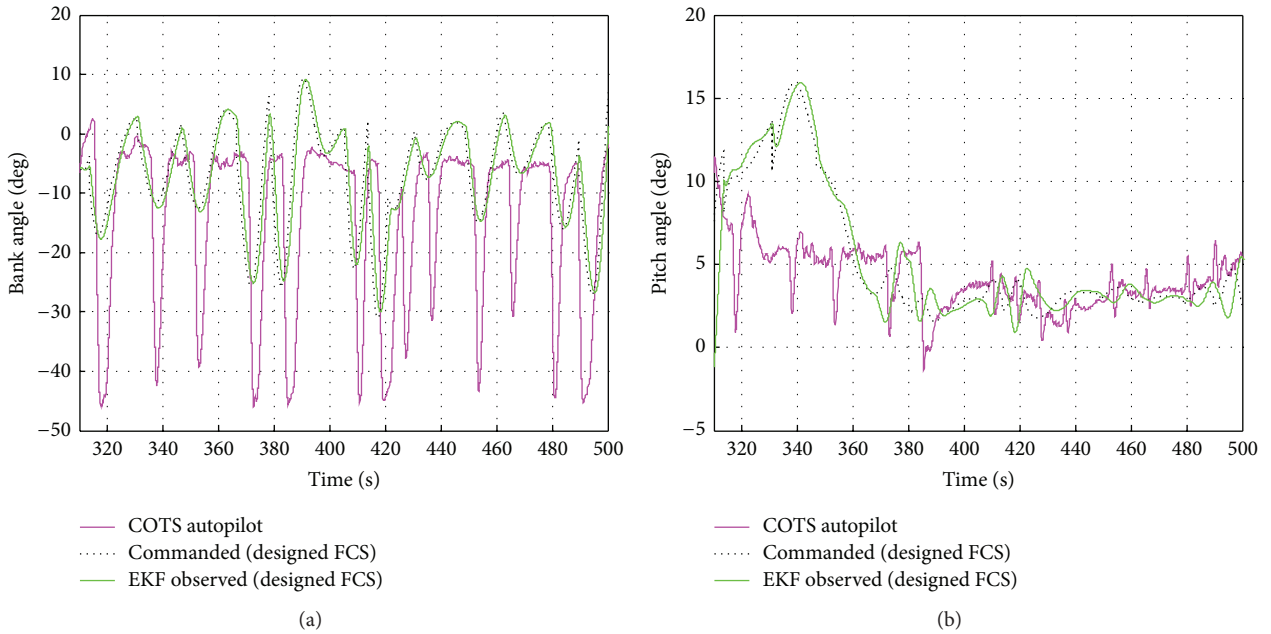


FIGURE 23: Bank angle (a). Pitch angle (b).

relation $\bar{\mathbf{D}}_{21}^i \bar{\mathbf{D}}_{21}^{i T} = \mathbf{I} - \bar{\mathbf{D}}_{1112}^i ((\gamma^i)^2 \mathbf{I} - \bar{\mathbf{D}}_{1111}^{i T} \bar{\mathbf{D}}_{1111}^i)^{-1} \bar{\mathbf{D}}_{1112}^{i T}$.
The feedback \mathbf{F}^i and state feedback \mathbf{H}^i matrices are

$$\mathbf{F}^i = \begin{bmatrix} \mathbf{F}_{11}^i \\ \mathbf{F}_{12}^i \\ \mathbf{F}_2^i \end{bmatrix} \begin{matrix} \{m_1 - p_2\} \times \{n\} \\ \{p_2\} \times \{n\} \\ \{m_2\} \times \{n\} \end{matrix}$$

$$= -(\mathbf{R}_a^i)^{-1} [\mathbf{D}_{1a}^{i T} \mathbf{C}_{1i}^i + \mathbf{B}^{i T} \mathbf{X}^i],$$

$$\mathbf{H}^i = \begin{bmatrix} \mathbf{H}_{11}^i & \mathbf{H}_{12}^i & \mathbf{H}_2^i \\ (n) \times (p_1 - m_2) & (n) \times (m_2) & (n) \times (p_2) \end{bmatrix}$$

$$= -[\mathbf{B}^i \mathbf{D}_{1b}^{i T} + \mathbf{Y}^i \mathbf{C}^{i T}] (\mathbf{R}_b^i)^{-1}, \quad (\text{D.5})$$

where

$$\mathbf{B}^i = [\mathbf{B}_1^i \quad \mathbf{B}_2^i], \quad \mathbf{C}^i = \begin{bmatrix} \mathbf{C}_1^i \\ \mathbf{C}_2^i \end{bmatrix},$$

$$\mathbf{Z}^i = (\mathbf{I} - (\gamma^i)^{-2} \mathbf{Y}^i \cdot \mathbf{X}^i)^{-1},$$

$$\begin{aligned} \mathbf{D}_{1a}^i &= [\mathbf{D}_{11}^i \quad \mathbf{D}_{12}^i], & \mathbf{D}_{1b}^i &= \begin{bmatrix} \mathbf{D}_{11}^i \\ \mathbf{D}_{21}^i \end{bmatrix}, \\ \mathbf{R}_a^i &= \mathbf{D}_{1a}^{i T} \mathbf{D}_{1a}^i - \begin{bmatrix} (\gamma^i)^2 \mathbf{I}_{m_1} & 0 \\ 0 & 0 \end{bmatrix}, & (D.6) \\ \mathbf{R}_b^i &= \mathbf{D}_{1b}^i \mathbf{D}_{1b}^{i T} - \begin{bmatrix} (\gamma^i)^2 \mathbf{I}_{p_1} & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

Matrices \mathbf{X}^i and \mathbf{Y}^i are the solutions of the following Riccati equations:

$$\begin{aligned} & (\mathbf{A}^i - \mathbf{B}^i (\mathbf{R}_a^i)^{-1} \mathbf{D}_{1a}^{i T} \mathbf{C}_1^i)^T \mathbf{X}^i + \mathbf{X}^i (\mathbf{A}^i - \mathbf{B}^i (\mathbf{R}_a^i)^{-1} \mathbf{D}_{1a}^{i T} \mathbf{C}_1^i) \\ & - \mathbf{X}^i (\mathbf{B}^i (\mathbf{R}_a^i)^{-1} \mathbf{B}^{i T}) \mathbf{X}^i + \mathbf{C}_1^{i T} (\mathbf{I} - \mathbf{D}_{1a}^i (\mathbf{R}_a^i)^{-1} \mathbf{D}_{1a}^{i T}) \mathbf{C}_1^i \\ & = 0, \\ & (\mathbf{A}^i - \mathbf{B}_1^i \mathbf{D}_{1b}^{i T} (\mathbf{R}_b^i)^{-1} \mathbf{C}_1^i) \mathbf{Y}^i + \mathbf{Y}^i (\mathbf{A}^i - \mathbf{B}_1^i \mathbf{D}_{1b}^{i T} (\mathbf{R}_b^i)^{-1} \mathbf{C}_1^i)^T \\ & - \mathbf{Y}^i (\mathbf{C}_1^{i T} (\mathbf{R}_b^i)^{-1} \mathbf{C}_1^i) \mathbf{Y}^i + \mathbf{B}_1^i (\mathbf{I} - \mathbf{D}_{1b}^i \mathbf{D}_{1b}^{i T} (\mathbf{R}_b^i)^{-1} \mathbf{D}_{1b}^i) \mathbf{B}_1^{i T} \\ & = 0. \end{aligned} \quad (D.7)$$

No new information is considered in the prediction stage, and the one-step ahead prediction state vector $\hat{\mathbf{x}}[k/k-1]$ is obtained by propagating the equation $\hat{\mathbf{f}}$ from previous state and input vectors at time $[k-1]$. The correction phase delivers $\hat{\mathbf{x}}[k/k]$ by adjusting $\hat{\mathbf{x}}[k/k-1]$ based on the difference between measurements and prediction, that is, $\mathbf{y}[k] - \hat{\mathbf{h}}(\hat{\mathbf{x}}[k/k-1], \hat{\mathbf{u}}[k])$, through the Kalman gain $\mathbf{K}[k]$. In (E.2), \mathbf{P} acts as the state error $(\mathbf{x} - \hat{\mathbf{x}})$ covariance, and matrices \mathbf{F} and \mathbf{H} constitute the Jacobian of the propagation equation $\hat{\mathbf{f}}$ and observation equation $\hat{\mathbf{h}}$, respectively; that is, $\tilde{\mathbf{F}}[k-1] = \partial \hat{\mathbf{f}} / \partial \mathbf{x}_{\hat{\mathbf{x}}[k-1/k-1], \mathbf{u}[k-1]}$ and $\tilde{\mathbf{H}}[k] = \partial \hat{\mathbf{h}} / \partial \mathbf{x}_{\hat{\mathbf{x}}[k/k-1]}$. $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{R}}$ are the covariance matrices of \mathbf{v} and \mathbf{w} , respectively.

Acknowledgment

This research was supported by the NSF Center for Remote Sensing of Ice Sheet (CReSIS) Grant NSF-0066685, at the University of Kansas.

E. Extended Kalman Filter Algorithm

The filter's functions $\hat{\mathbf{f}}$ and $\hat{\mathbf{h}}$ represent the observed system. They incorporate all available information of the aircraft characterized by functions \mathbf{f} and \mathbf{h} and other variables to be estimated as biases and wind; that is, $\hat{\mathbf{f}} \supseteq \mathbf{f}$ and $\hat{\mathbf{h}} \supseteq \mathbf{h}$. Model mismatch $\mathbf{v}(t)$ and sensor noise $\mathbf{w}(t)$ are included through the Kalman gain $\mathbf{K}(t)$ calculation as shown later. The continuous-time observer is defined as

$$\begin{aligned} \dot{\hat{\mathbf{x}}}(t) &= \tilde{\mathbf{f}}(\hat{\mathbf{x}}(t), \mathbf{u}(t)) + \tilde{\mathbf{K}}(t) (\mathbf{y}(t) - \hat{\mathbf{y}}(t)), \\ \hat{\mathbf{y}}(t) &= \tilde{\mathbf{h}}(\hat{\mathbf{x}}(t)), \end{aligned} \quad (E.1)$$

where $\hat{\mathbf{x}}(t) \in R^{\hat{n}}$ is the filter state vector and $\hat{\mathbf{u}}(t) \in R^{\hat{m}}$ is the filter input vector.

Aircraft dynamics and observation equations $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{v}(t)$ and $\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(t)$ are assumed to be additively corrupted by uncorrelated Gaussian noise signals \mathbf{v} and \mathbf{w} . Applying the Euler discretization method, the \hat{n} order discrete version is obtained $\hat{\mathbf{x}}[k+1] = \hat{\mathbf{x}}[k] + \Delta T \tilde{\mathbf{f}}(\hat{\mathbf{x}}[k], \hat{\mathbf{u}}[k])$ and $\hat{\mathbf{y}}[k] = \tilde{\mathbf{h}}(\hat{\mathbf{x}}[k], \hat{\mathbf{u}}[k])$, valid at each sampling time interval ΔT , where index $[k]$ corresponds to time (t) sampled at $t = k\Delta T$. The discrete extended Kalman filter algorithm is defined in two phases:

$$\begin{aligned} \text{prediction} & \begin{cases} \hat{\mathbf{x}}[k/k-1] = \hat{\mathbf{f}}(\hat{\mathbf{x}}[k-1/k-1], \hat{\mathbf{u}}[k-1]), \\ \mathbf{P}[k/k-1] = \mathbf{F}[k-1] \mathbf{P}[k-1/k-1] \mathbf{F}^T[k-1] + \mathbf{Q}[k-1], \end{cases} \\ \text{correction} & \begin{cases} \mathbf{K}[k] = \mathbf{P}[k/k-1] \mathbf{H}^T[k] (\mathbf{H}[k] \mathbf{P}[k/k-1] \mathbf{H}^T[k] + \mathbf{R}[k])^{-1}, \\ \hat{\mathbf{x}}[k/k] = \hat{\mathbf{x}}[k/k-1] + \mathbf{K}[k] (\mathbf{y}[k] - \hat{\mathbf{h}}(\hat{\mathbf{x}}[k/k-1], \hat{\mathbf{u}}[k])), \\ \mathbf{P}[k/k] = [\mathbf{I} - \mathbf{K}[k] \mathbf{H}[k]] \mathbf{P}[k/k-1]. \end{cases} \end{aligned} \quad (E.2)$$

References

- [1] I. Kammer, A. Pascoal, E. Hallberg, and C. Silvestre, "Trajectory tracking for autonomous vehicles: an integrated approach to guidance and control," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 29–38, 1998.
- [2] M. Niculescu, "Lateral track control law for Aerosonde UAV," in *Proceedings of the 39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nev, USA, 2001.
- [3] D. Boyle and G. Chamitoff, "Robust nonlinear lasso control: a new approach for autonomous trajectory tracking," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Tex, USA, 2003.
- [4] S. Park, J. Deyst, and J. How, "A new nonlinear guidance logic for trajectory tracking," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, RI, USA, 2004.
- [5] J. Osborne and R. Rysdyk, "Waypoint guidance for small UAVs in wind," in *Proceedings of the AIAA Infotech, Aerospace Conference*, Arlington, Va, USA, 2005.

- [6] R. Rysdyk, "Unmanned aerial vehicle path following for target observation in wind," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1092–1100, 2006.
- [7] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.
- [8] L. Sonneveldt, E. R. Van Oort, Q. P. Chu, and J. A. Mulder, "Nonlinear adaptive trajectory control applied to an F-16 model," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 25–39, 2009.
- [9] G. Garcia, S. Keshmiri, and R. Colgren, "Advanced H-infinity trainer autopilot," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, Toronto, Canada, 2010.
- [10] J. A. Rios and E. White, "Fusion filter algorithm enhancements for a MEMS GPS/IMU," in *Proceedings of the 14th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS '01)*, pp. 1382–1393, Salt Lake City, Utah, USA, September 2001.
- [11] P. Zhang, J. Gu, E. E. Milios, and P. Huynh, "Navigation with IMU/GPS/digital compass with unscented Kalman filter," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '05)*, pp. 1497–1502, Niagara Falls, Canada, August 2005.
- [12] Y. Li, J. Wang, C. Rizos, P. Mumford, and W. Ding, "Low-cost tightly coupled GPS/INS integration based on a nonlinear Kalman filtering design," in *Proceedings of the Institute of Navigation, National Technical Meeting (NTM '06)*, pp. 958–966, Monterey, Calif, USA, January 2006.
- [13] X. Lin-lin, W. Jian-guo, Z. Li-hui, and G. Li-jun, "Nonlinear gaussian filter algorithm enhancements for low-cost integrated navigations systems," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp. 4566–4571, Changchun, China, August 2009.
- [14] Y. Ren and X. Ke, "Particle filter data fusion enhancements for MEMS-IMU/GPS," *Intelligent Information Management*, vol. 2, pp. 417–421, 2010.
- [15] R. Van der Merwe and E. Wan, "Sigma-point kalman filters for integrated navigation," in *Proceedings of the 60th Annual Meeting of the Institute of Navigation (ION '04)*, Dayton, Ohio, USA, 2004.
- [16] A. Nemra and N. Aouf, "Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering," *IEEE Sensors Journal*, vol. 10, no. 4, pp. 789–798, 2010.
- [17] A. Eldredge, *Improved state estimation for miniature air vehicles [M.S. thesis]*, Department of Mechanical Engineering, Brigham Young University, Provo, Utah, USA, 2006.
- [18] M. Pachter, N. Ceccarelli, and P. R. Chandler, "Estimating MAV's heading and the wind speed and direction using GPS, inertial, and air speed measurements," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, USA, August 2008.
- [19] D. J. Linse and R. F. Stengel, "Identification of aerodynamic coefficients using computational neural networks," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 6, pp. 1018–1025, 1993.
- [20] W. E. Faller and S. J. Schreck, "Real-time prediction of unsteady aerodynamics: application for aircraft control and maneuverability enhancement," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1461–1468, 1995.
- [21] M. Larson, P. De Raedt, and M. Hedlund, "Aerodynamic identification using neural networks," Reptort LiTH-154-1937, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1997.
- [22] S. C. Raisinghani, G. V. S. Reddy, and A. K. Ghosh, "Parameter estimation of an augmented aircraft using neural networks," in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Portland, Ore, USA, 1999.
- [23] S. C. Raisinghani and A. K. Ghosh, "Parameter estimation of an unstable aircraft in turbulent atmosphere using neural networks," in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Monterey, Calif, USA, 2002.
- [24] N. K. Peyada and A. K. Ghosh, "Aircraft parameter estimation using a new filtering technique based upon a neural network and Gauss-Newton method," *Aeronautical Journal*, vol. 113, no. 1142, pp. 243–252, 2009.
- [25] S. Das, R. A. Kuttieri, M. Sinha, and R. Jategaonkar, "Neural partial differential method for extracting aerodynamic derivatives from flight data," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 376–384, 2010.
- [26] Z. Wang, C. E. Lan, and J. M. Brandon, "Fuzzy logic modeling of nonlinear unsteady aerodynamics," in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Boston, Mass, USA, 1998.
- [27] Z. Wang, C. E. Lan, and J. M. Brandon, "Fuzzy logic modeling of lateral-directional unsteady aerodynamics," in *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Boston, Mass, USA, 1998.
- [28] A. Hossain, A. Rahman, J. Hossen, A. K. M. P. Iqbal, and M. I. Zahirul, "Prediction of aerodynamic characteristics of an aircraft model with and without winglet using fuzzy logic technique," *Aerospace Science and Technology*, vol. 15, no. 8, pp. 595–605, 2011.
- [29] C. E. Lan, S. Keshmiri, and R. Hale, "Fuzzy-logic modeling of a rolling unmanned vehicle in antarctica wind shear," *AIAA Journal of Guidance, Control and Dynamics*, vol. 35, no. 5, pp. 1538–1547, 2012.
- [30] B. Stevens and F. Lewis, *Aircraft Control and Simulation*, John Wiley & Sons, Hoboken, NJ, USA, 2nd edition, 2003.
- [31] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls (Part I)*, DARcorporation, Lawrence, Kan, USA, 2003.
- [32] DARcorporation, *Advanced Aircraft Analysis*, Software Package, Wichita, Kan, USA, 2009.
- [33] E. Leong, *Development of A 6DOF Nonlinear Simulation Model Enhanced with Fine Tuning Procedures*, The University of Kansas, 2008.
- [34] K. Glover and J. C. Doyle, "State-space formulae for all stabilizing controllers that satisfy an H_{∞} -norm bound and relations to relations to risk sensitivity," *Systems and Control Letters*, vol. 11, no. 3, pp. 167–172, 1988.
- [35] K. Zhou and J. Doyle, *Essentials of Robust Control*, Prentice Hall, Upper Saddle River, NJ, USA, 1998.
- [36] G. Garcia, S. Keshmiri, and R. Colgren, "H-infinity gain scheduling design for the Meridian UAS for a broader range of operation and for fault tolerant applications," in *Proceedings of the 9th IEEE International Conference on Control and Automation (ICCA '11)*, pp. 1174–1180, December 2011.
- [37] S. Park, J. Deyst, and J. P. How, "Performance and lyapunov stability of a nonlinear path-following guidance method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.
- [38] D. Simon, *Optimal State Estimation*, John Wiley & Sons, Hoboken, NJ, USA, 2006.

- [39] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [40] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, 1999.



Hindawi
Submit your manuscripts at
<http://www.hindawi.com>

