JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 25, No. 3, May–June 2002

Online Parameter Estimation Techniques Comparison Within a Fault Tolerant Flight Control System

Yongkyu Song*
Hankuk Aviation University, Kyonggido 412-791, Republic of Korea
and
Giampiero Campa,† Marcello Napolitano,‡ Brad Seanor,§ and Mario G. Perhinschi†
West Virginia University, Morgantown, West Virginia 26506-6106

The results of a study where two online parameter identification (PID) methods are compared for application within a fault tolerant flight control system are described. One of the PID techniques is time-domain based, whereas the second is featured in the frequency domain. The time-domain method was directly suitable for the online estimates of the dimensionless aircraft stability derivatives. The frequency-domain method was modified from its original formulation to provide direct estimates of the stability derivatives. This effort was conducted within the research activities of the NASA Intelligent Flight Control System F-15 program. The comparison is performed through dynamic simulations with a specific procedure to model the aircraft aerodynamics following the occurrence of a battle damage/failure on a primary control surface. The two PID methods show similar performance in terms of accuracy of the estimates, convergence time, and robustness to noise. However, the frequency-domain-based method outperforms the time-domain-based method in terms of computational requirements for online real-time applications. The study has also emphasized the advantages of using ad hoc short preprogrammed maneuvers to provide enough excitation following the occurrence of the actuator failure to allow the parameter estimation process.

Nomenclature

c	=	aerodynamic coefficient
K	=	numerical coefficient
L	=	lift force
q	=	pitch angular velocity, rad/s
S	=	wing surface, ft^2
t	=	time, s
x	=	longitudinal axis
y	=	lateral axis
z	=	vertical axis
α	=	angle of attack, rad or deg
δ	=	control surface deflection, rad or deg
ε	=	downwash angle, rad or deg
η	=	dynamic pressure ratio
θ	=	pitch Euler angle, rad or deg
σ	=	standard deviation
ω	=	frequency, rad/s

Subscripts

h	=	horizontal tail
L	=	left side
l	=	rolling moment
M	=	pitching moment
R	=	right side
S	=	stabilator

Received 11 May 2000; revision received 10 July 2001; accepted for publication 22 October 2001.

*Associate Professor, School of Aerospace and Mechanical Engineering, 200-1 Hwajondong, Goyangshi. Member AIAA.
†Research Assistant Professor, Department of Mechanical and Aerospace Engineering. Member AIAA.
‡Professor, Department of Mechanical and Aerospace Engineering.
§Graduate Research Assistant, Department of Mechanical and Aerospace Engineering. Member AIAA.

Introduction

AIRCRAFT parameter estimation from flight data has been extensively conducted as a postflight analysis for several years. Several statistical methods have been used for this purpose, with the maximum likelihood method one of the most widely used approaches.[1–3]

In recent years, drastic increases in the available onboard computational power have allowed the flight control community to consider the application of online parameter estimation techniques. In particular, the online extension of the parameter identification (PID) process has immediate and potentially very important applications for control of time-varying aircraft systems, such as an aircraft subjected to substantial changes in the dynamic and aerodynamic characteristics.

On a parallel path, research on fault-tolerant flight control systems has been an important issue in flight controls. A fault-tolerant flight control system is required to perform failure detection, identification, and accommodation following a battle damage and/or failure to a critical control surface. To implement a failure accommodation strategy, a variety of control surfaces (speed brakes, wing flaps, differential dihedral canards, spoilers, etc.) and thrust mechanisms (differential thrust, thrust vectoring) can be used.[4]

Recent experimental research programs, such as the Self-Designing Controller (SDC) program,[5] the RESTORE program,[6,7] and the Intelligent Flight Control System (IFCS) F-15 program (previously known as the F-15 ACTIVE program),[8,9] have proposed specific fault-tolerant control laws formulated using online estimates of aircraft parameters obtained from a real time PID scheme.

The research effort described in this study was conducted within the NASA IFCS F-15 program. A block diagram of the IFCS F-15 fault-tolerant scheme is shown in Fig. 1 (Ref. 9). The scheme features a set of baseline neural networks acting as look-up tables for 26 stability and control derivatives (SCDs) of the IFCS F-15. At nominal flight conditions, these 26 SCDs are provided as inputs to the IFCS F-15 controller featuring control laws designed through online solution of the Riccati equations. Following a failure on a primary control surface, a PID scheme is tasked with providing online estimates of the same 26 SCDs to an online neural network (OLNN). The OLNN's task is to interpolate throughout the flight IFCS F-15 flight envelope the correction terms for the 26 SCDs, which, added to the values from the baseline neural networks, are provided as inputs to the controller. The current effort

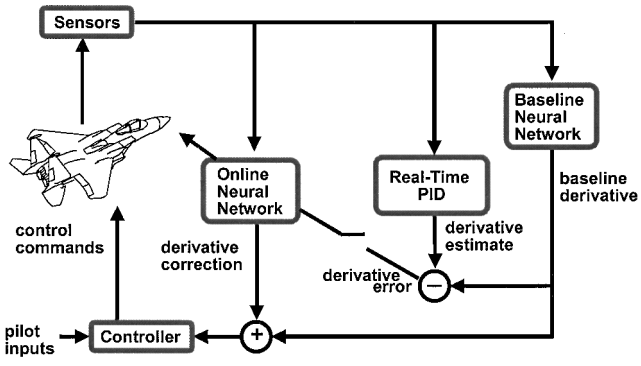

Fig. 1 Fault tolerant scheme for the NASA IFCS F-15.

focuses on the selection of a suitable method for the online PID task.

Like offline PID approaches, online PID methods can be formulated either in the time domain or in the frequency domain. Within time-domain online PID techniques least squares (LS) algorithms are used in lieu of techniques based on the use of the gradient and Hessian because of their lower computational effort and better convergence characteristics. Therefore, online time-domain PID techniques mainly include variations of the LS regression method, such as recursive LS (RLS),[10,11] RLS with a forgetting factor,[12] a modified sequential LS,[5] a real-time batch LS,[13,14] and extended Kalman filtering.[15]

The real-time application of any of these methods is challenging due to a combination of the unavoidable presence of system and measurement noise, possible lack of information for PID purposes in the flight data (such as a prolonged steady state flight condition), and potential unavailability of independent control inputs, a necessary condition for an accurate PID, due to the interactions with the closed-loop control laws. Analytical mechanisms to handle some of these problems include the use of temporal and spatial constraints (such as forgetting factors and/or the use of short sets of flight data). A potential problem with the previous time-domain PID techniques is the lack of a reliable parameter for an online assessment of the accuracy of the estimates in the presence of unmodeled noise. Another less known problem for real-time application of PID techniques is the presence of time skews and/or delays from the instant flight data are acquired by the flight control system and the instant the data are supplied to the PID algorithm.

To overcome some or all of the problems described, two techniques, one time-domain based and one frequency-domain based, have recently been introduced. The primary objective of this effort was to conduct a detailed comparison of the performance of these two PID techniques through simulations with a detailed aerodynamic modeling. Because of the real-time nature of the application of the PID process, the comparison was performed using the following parameters: 1) convergence time for the parameters to be estimated, 2) CPU requirements, 3) robustness to measurment and system noise, 4) amount of excitation required for an effective estimation, and 5) online computed variance of the estimation error.

The paper is organized as follows. The next section briefly reviews a detailed approach of the aerodynamic modeling for an aircraft at postfailure and/or damage conditions. Two following sections review the two online PID methods. Then a section discusses the comparison of the two methods following control surface damages typical of a combat situation for a military aircraft under different scenarios. A final section summarizes the paper with conclusions.

Aerodynamic Modeling of Actuator Failure and/or Battle Damage

In a worst-case scenario, actuator failure and/or battle damage may imply a missing surface or, more realistically, a missing surface with the actuator jammed at a given position. When a simplified modeling is used, failure and/or battle failure can be modeled as jammed actuators without missing surface. In this effort, emphasis has been placed in the accurate postfailure aerodynamic model for simulation purposes.

From conventional aerodynamic modeling,[16,17] the aerodynamic characteristics of a surface can be expressed in terms of normal force, axial force, and moment force around some fixed points or axes. In this paper, we will examine failures in the longitudinal axis only. This failure is believed to be more critical than aileron and rudder failures due to the coupling between the longitudinal and lateral–directional dynamics.

A control surface damage with a missing portion induces instantaneous changes in its aerodynamic characteristics. An assumption in the aerodynamic modeling is that the axial forces exerted by a control surface deflection (in this case longitudinal surfaces) are negligible. Thus, the net effect of any control surface failure with a missing portion is a change (a reduction) in the relative normal force coefficient. The aerodynamic moments around the different axes can then be considered proportional to this normal force coefficient through the aircraft geometric parameters. Therefore, to test the capabilities of the PID schemes to evaluate online the postfailure mathematical model, the objective is to obtain closed-form expressions of the nondimensional aerodynamic stability and control derivatives as a function of the normal force coefficient relative to the control surface object of failure and/or battle damage. When conventional aerodynamic modeling is used, these closed-form relationships can be obtained for conventional subsonic aerodynamic configurations. A more detailed effort involving higher level modeling would be required for supersonic conditions.

The described aerodynamic modeling was performed using the mathematical model for an F-4 aircraft[16] at subsonic flight conditions. Because a simulation code for the IFCS F-15 was not readily available, the F-4 model was selected because of general similarities with the F-15 model in the size of the control surfaces. A simulation code was built using linearized aerodynamics and nonlinear dynamic equations. F-4 data are summarized in Table 1. The SCDs are as follows:

$$c_{L_1} = 0.26, \quad c_{D_1} = 0.03, \quad c_{T_{X_1}} = 0.03, \quad c_{m_1} = 0.0$$

$$c_{m_{T_1}} = 0.0, \quad c_{D_0} = 0.0205, \quad c_{D_u} = 0.027, \quad c_{D_\alpha} = 0.3$$

$$c_{D_{\delta S}} = -0.1, \quad c_{T_{X_u}} = -0.064, \quad c_{L_0} = 0.1$$

$$c_{L_u} = 0.27, \quad c_{L_\alpha} = 3.75, \quad c_{L_{\dot\alpha}} = 0.86, \quad c_{L_q} = 1.8$$

$$c_{L_{\delta S}} = 0.4, \quad c_{m_0} = 0.025, \quad c_{m_u} = -0.117$$

$$c_{m_\alpha} = -0.4, \quad c_{m_{\dot\alpha}} = -1.3, \quad c_{m_q} = -2.7$$

$$c_{m_{T_u}} = c_{m_{T_\alpha}} = 0, \quad c_{m_{\delta S}} = -0.58, \quad c_{l_\beta} = -0.08$$

$$c_{l_p} = -0.24, \quad c_{l_r} = 0.07, \quad c_{l_{\delta A}} = 0.042, \quad c_{l_{\delta R}} = 0.006$$

$$c_{y_\beta} = -0.68, \quad c_{y_p} = 0.0, \quad c_{y_r} = 0.0, \quad c_{y_{\delta A}} = -0.016$$

Table 1 F-4 aircraft data (subsonic cruise condition)

Property	Value
Flight conditions	
Altitude, ft	35,000
Mach number	0.9
Air speed, ft/s	876
Dynamic pressure, lb/ft^2	283.2
c.g. Location (chord fraction)	0.29
Angle of attack, deg	2.6
Reference geometry	
Wing surface, ft^2	530
Mean aerodynamic chord, ft	16
Wing span, ft	38.7
Inertial	
Weight, lb	39,000
I_{xx}, slug ft^2	25,000
I_{yy}, slug ft^2	122,200
I_{zz}, slug ft^2	139,800
I_{xz}, slug ft^2	2,200

$$c_{y_{\delta R}} = 0.095, \qquad c_{n_\beta} = 0.125, \qquad c_{n_p} = -0.036$$

$$c_{n_r} = -0.27, \qquad c_{n_{\delta A}} = -0.001, \qquad c_{n_{\delta R}} = -0.066$$

For fault tolerance and PID purposes, it is assumed that the F-4 has stabilators that can be decoupled as two independent surfaces at post-failure conditions. The objective is to obtain closed-form expressions for the following derivatives: c_{L_α}, c_{m_α}, $c_{L_{\dot\alpha}}$, $c_{m_{\dot\alpha}}$, c_{L_q}, and c_{m_q} in terms of the c_{L_δ} of the left and right side of the longitudinal control surface, in this case stabilators. A similar expression is also needed for the induced rolling moment, defined as $\Delta c_l|_{\text{failure}}$. When aerodynamic modeling is used for subsonic flight conditions,[16,17] it is known that

$$c_{L_\alpha} = c_{L_{\alpha WB}} + \eta_H \frac{S_H}{S}\left(1 - \frac{\partial\varepsilon}{\partial\alpha}\right)c_{L_\alpha H} \tag{1}$$

Also, for the stabilator, we would have

$$c_{L_{\delta_S}} = c_{L_{\delta S\_R}} + c_{L_{\delta_{S\_L}}} = \eta_H (S_H/S) c_{L_\alpha H} \Rightarrow c_{L_\alpha H}$$

$$= c_{L_{\delta_S}} \big/ \eta_H (S_H/S) \tag{2}$$

Therefore, breaking down the $c_{L_{\delta_S}}$ contribution from the left and right stabilator, we would have the following expression for c_{L_α}:

$$c_{L_\alpha} = c_{L_{\alpha WB}} + \left(1 - \frac{\partial\varepsilon}{\partial\alpha}\right)c_{L_{\delta S\_R}} + \left(1 - \frac{\partial\varepsilon}{\partial\alpha}\right)c_{L_{\delta S\_L}} \tag{3}$$

When aircraft geometric data[17] is used, a numerical value for the downwash effect was found for the given flight conditions. When such a value is used and with knowledge of the value for c_{L_α} at nominal conditions, it was possible to solve for a value for $c_{L_{\alpha WB}}$. Therefore, numerical values were found for the preceding c_{L_α} relationship leading to the expression

$$c_{L_\alpha} = K_{1,1} + K_{1,2} c_{L_{\delta S\_R}} + K_{1,3} c_{L_{\delta S\_L}} \tag{4}$$

where the numerical values for the K coefficients are reported as follows:

$$c_{L_\alpha} \Rightarrow K_{1,1} = 3.53, \qquad K_{1,2} = 0.55, \qquad K_{1,3} = 0.55$$

$$c_{m_\alpha} \Rightarrow K_{2,1} = 0.07, \qquad K_{2,2} = -1.17, \qquad K_{2,3} = -1.17$$

$$c_{L_{\dot\alpha}} \Rightarrow K_{3,1} = 0.093, \qquad K_{3,2} = 1.917, \qquad K_{3,3} = 1.917$$

$$c_{m_{\dot\alpha}} \Rightarrow K_{4,1} = 0.333, \qquad K_{4,2} = -4.083, \qquad K_{4,3} = -4.083$$

$$c_{L_q} \Rightarrow K_{5,1} = 0.096, \qquad K_{5,2} = 4.26, \qquad K_{5,3} = 4.26$$

$$c_{m_q} \Rightarrow K_{6,1} = 1.292, \qquad K_{6,2} = -9.982, \qquad K_{6,3} = -9.982$$

$$\Delta c_\ell|_{\text{failure}} \Rightarrow \left(\bar y_{c_H}/b/2\right) = 0.023$$

A similar approach was used for the modeling of c_{m_α}. Starting from the following expression:

$$c_{m_\alpha} = c_{m_{\alpha WB}} + c_{m_{\alpha H}} = c_{L_{\alpha WB}}\left(\bar x_{\text{cg}} - \bar x_{AC_{WB}}\right)$$

$$- \eta_H \frac{S_H}{S}\left(1 - \frac{\partial\varepsilon}{\partial\alpha}\right)\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)c_{L_\alpha H} \tag{5}$$

and using $c_{L_\alpha H} = c_{L_{\delta_S}} S/S_H \eta_H$, we would have

$$c_{m_\alpha} = c_{m_{\alpha WB}} - \left(1 - \frac{\partial\varepsilon}{\partial\alpha}\right)\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)c_{L_{\delta S\_R}}$$

$$- \left(1 - \frac{\partial\varepsilon}{\partial\alpha}\right)\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)c_{L_{\delta S\_L}} \tag{6}$$

When $\bar x_{\text{cg}}$ was known from the available data[16] and $(\bar x_{AC_H} - \bar x_{\text{cg}})$ was evaluated from the aircraft geometry, it was possible to solve for $c_{m_{\alpha WB}}$ starting from the nominal value for c_{m_α}. Thus, numerical values were found for the preceding c_{m_α} relationship leading to the expression

$$c_{m_\alpha} = K_{2,1} + K_{2,2} c_{L_{\delta S\_R}} + K_{2,3} c_{L_{\delta S\_L}} \tag{7}$$

where the numerical values for the K coefficients were given earlier. Next the modeling of $c_{L_{\dot\alpha}}$ was performed starting from the following expression:

$$c_{L_{\dot\alpha}} = c_{L_{\dot\alpha WB}} + c_{L_{\dot\alpha H}} = c_{L_{\dot\alpha WB}} + 2c_{L_{\alpha H}}\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)\eta_H \frac{S_H}{S}\left(\frac{\partial\varepsilon}{\partial\alpha}\right) \tag{8}$$

Using $c_{L_{\alpha H}} = c_{L_{\delta_S}} S/S_H \eta_H$, we would have

$$c_{L_{\dot\alpha}} = c_{L_{\dot\alpha WB}} + 2\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)\left(\frac{\partial\varepsilon}{\partial\alpha}\right)c_{L_{\delta S\_R}}$$

$$+ 2\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)\left(\frac{\partial\varepsilon}{\partial\alpha}\right)c_{L_{\delta S\_L}} \tag{9}$$

With the values for $(\bar x_{AC_H} - \bar x_{\text{cg}})$, $(\partial\varepsilon/\partial\alpha)$ already determined, the value for $c_{L_{\dot\alpha WB}}$ can be found using the provided nominal value for $c_{L_{\dot\alpha}}$. As before, values were found for the following $c_{L_{\dot\alpha}}$ relationship:

$$c_{L_{\dot\alpha}} = K_{3,1} + K_{3,2} c_{L_{\delta S\_R}} + K_{3,3} c_{L_{\delta S\_L}} \tag{10}$$

where the numerical values for the K coefficients were given earlier. Next, the derivative $c_{m_{\dot\alpha}}$ was analyzed. Starting from the expression

$$c_{m_{\dot\alpha}} = c_{m_{\dot\alpha WB}} + c_{m_{\dot\alpha H}} = c_{m_{\dot\alpha WB}} - 2c_{L_{\alpha H}}\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)^2 \eta_H \frac{S_H}{S}\left(\frac{\partial\varepsilon}{\partial\alpha}\right) \tag{11}$$

using $c_{L_{\alpha H}} = c_{L_{\delta S}} S/S_H \eta_H$, we would have

$$c_{m_{\dot\alpha}} = c_{m_{\dot\alpha WB}} - 2\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)^2\left(\frac{\partial\varepsilon}{\partial\alpha}\right)c_{L_{\delta S\_R}}$$

$$- 2\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)^2\left(\frac{\partial\varepsilon}{\partial\alpha}\right)c_{L_{\delta S\_L}} \tag{12}$$

The value for $c_{m_{\dot\alpha WB}}$ is found using the provided nominal $c_{m_{\dot\alpha}}$ leading to

$$c_{m_{\dot\alpha}} = K_{4,1} + K_{4,2} c_{L_{\delta S\_R}} + K_{4,3} c_{L_{\delta S\_L}} \tag{13}$$

The same approach is used for the remaining longitudinal derivatives c_{L_q} and c_{m_q}. For c_{L_q}, starting from

$$c_{L_q} = c_{L_{qWB}} + c_{L_{qH}} = c_{L_{qWB}} + 2c_{L_{\alpha H}}\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)\eta_H (S_H/S)$$

$$= c_{L_{qWB}} + 2\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)c_{L_{\delta S\_R}} + 2\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)c_{L_{\delta S\_L}} \tag{14}$$

With the value for $c_{L_{qWB}}$ found using the nominal value for c_{L_q}, the following expression is achieved:

$$c_{L_q} = K_{5,1} + K_{5,2} c_{L_{\delta S\_R}} + K_{5,3} c_{L_{\delta S\_L}} \tag{15}$$

Similarly, for c_{m_q}, starting from

$$c_{m_q} = c_{m_{qWB}} + c_{m_{qH}} = c_{m_{qWB}} - 2.2c_{L_{\alpha H}}\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)^2 \eta_H (S_H/S)$$

$$= c_{m_{qWB}} - 2.2\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)^2 c_{L_{\delta S\_R}} - 2.2\left(\bar x_{AC_H} - \bar x_{\text{cg}}\right)^2 c_{L_{\delta S\_L}} \tag{16}$$

With the value for $c_{m_{qWB}}$ found using the nominal value for c_{m_q}, the following expression is determined:

$$c_{m_q} = K_{6,1} + K_{6,2} c_{L_{\delta S\_R}} + K_{6,3} c_{L_{\delta S\_L}} \tag{17}$$

The preceding longitudinal modeling of the postfailure aerodynamics has been performed along the aircraft stability (aerodynamic) axes. With respect to the body axes, we would have instead

$$C_X = C_L \sin\alpha - C_D \cos\alpha, \qquad C_Z = -C_L \cos\alpha - C_D \sin\alpha \tag{18}$$

To complete the aerodynamic modeling, a coefficient is introduced to model the failure-induced rolling moment. With a standard sign convention for the rolling moment and the stabilator deflection used, the following coefficient is introduced:

$$\Delta c_\ell|_{\text{failure}} = -c_{L_{\delta S\_R}}(\bar{y}_{c_H}/b/2)\delta_{S\_R} + c_{L_{\delta S\_L}}(\bar{y}_{c_H}/b/2)\delta_{S\_L} \quad (19)$$

where $\bar{y}_{c_H}$ is the lateral location of the tail mean aerodynamic chord.

Frequency-Domain-Based Method for Online PID

The first online PID method in this study is based in the frequency domain and features a single-step technique based on discrete-time Fourier transform (DTFT).[11] This PID technique was originally introduced in Refs. 18–20 using previous work described in Ref. 21. The method has been substantially modified in this effort to use output equations, in lieu of state equations, and to provide direct estimates of the dimensionless stability derivatives in the body axes. For the F-4 aircraft dynamics,[16] the linearized equations are given by

$$\Delta T - m\Delta a_x = \bar{q}S\{C_{X_\alpha}\Delta\alpha + C_{X_q}(\bar{c}/2V)\Delta q + C_{X_u}(\Delta u/V) + C_{X_{\delta s\_R}}\Delta\delta_{S\_R} + C_{X_{\delta s\_L}}\Delta\delta_{S\_L}\} \quad (20)$$

$$m\Delta a_z = \bar{q}S\{C_{Z_\alpha}\Delta\alpha + C_{Z_q}(\bar{c}/2V)\Delta q + C_{z_u}(\Delta u/V) + C_{Z_{\delta s\_R}}\Delta\delta_{S\_R} + C_{Z_{\delta s\_L}}\Delta\delta_{S\_L}\} \quad (21)$$

$$m\Delta a_y = \bar{q}S\{C_{Y_\beta}\Delta\beta + C_{Y_p}(b/2V)\Delta p + C_{Y_r}(r/2V)\Delta r + C_{Y_{\delta A}}\Delta\delta_A + C_{Y_{\delta R}}\Delta\delta_R\} \quad (22)$$

$$I_x\Delta\dot{p} - I_{xz}\Delta\dot{r} = \bar{q}Sb\{C_{l_\beta}\Delta\beta + C_{l_p}(b/2V)\Delta p + C_{l_r}(b/2V)\Delta r + C_{l_{\delta A}}\Delta\delta_A + C_{l_{\delta s\_R}}\Delta\delta_{S\_R} + C_{l_{\delta s\_L}}\Delta\delta_{S\_L}\} \quad (23)$$

$$I_z\Delta\dot{r} - I_{xz}\Delta\dot{p} = \bar{q}Sb\{C_{n_\beta}\Delta\beta + C_{n_p}(b/2V)\Delta p + C_{n_r}(b/2V)\Delta r + C_{n_{\delta A}}\Delta\delta_A + C_{n_{\delta R}}\Delta\delta_R\} \quad (24)$$

$$I_y\Delta\dot{q} = \bar{q}S\bar{c}\{C_{m_\alpha}\Delta\alpha + C_{m_q}(\bar{c}/2V)\Delta q + C_{m_u}(\Delta u/V) + C_{m_{\delta s\_R}}\Delta\delta_{S\_R} + C_{m_{\delta s\_L}}\Delta\delta_{S\_L}\} \quad (25)$$

A general form for each of the preceding equations is given by

$$\boldsymbol{E}\dot{z}(t) + \boldsymbol{F}z(t) = \boldsymbol{x}(t)^T\boldsymbol{\Theta} \quad (26)$$

where $\boldsymbol{E}$ and $\boldsymbol{F}$ are known constant vectors and $\boldsymbol{\Theta}$ is an unknown constant vector to be estimated. For example, for the pitching moment equation we would have

$$x = [\Delta\alpha, \Delta q, \Delta\delta_{S\_R}, \Delta\delta_{S\_L}]^T, \qquad \boldsymbol{E}z = I_y\Delta q, \qquad \boldsymbol{F}z = 0$$

Sampling and applying DTFT to the input and motion variables at time $t = i\Delta t$, we have

$$j\omega\boldsymbol{E}\tilde{\boldsymbol{z}}(\omega) + \boldsymbol{F}\tilde{\boldsymbol{z}}(\omega) = \tilde{\boldsymbol{x}}(\omega)^T\boldsymbol{\Theta} \quad (27)$$

where

$$\tilde{\boldsymbol{x}}(\omega) = \sum_{i=0}^{N-1}\boldsymbol{x}(i\Delta t)e^{-j\omega i\Delta t}, \qquad \tilde{\boldsymbol{z}}(\omega) = \sum_{i=0}^{N-1}\boldsymbol{z}(i\Delta t)e^{-j\omega i\Delta t} \quad (28)$$

In this effort, Δt was set at 0.025 s. Also, the Fourier transform regression (FTR) was set up to analyze at each step 5 s. of data; thus, N is defined to be 200. As in the general LS regression method, the measurements of the vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ can be used to set up a cost function having the coefficients of $\boldsymbol{\Theta}$ as an argument. In particular, one can set m algebraic equations over a set of frequency points $[\omega_1, \omega_2, \ldots, \omega_m]$,

$$\begin{bmatrix} j\omega_1\boldsymbol{E}\tilde{y}(\omega_1) + \boldsymbol{F}\tilde{\boldsymbol{z}}(\omega_1) \\ j\omega_2\boldsymbol{E}\tilde{y}(\omega_2) + \boldsymbol{F}\tilde{\boldsymbol{z}}(\omega_2) \\ \cdots \\ \cdots \\ j\omega_m\boldsymbol{E}\tilde{y}(\omega_m) + \boldsymbol{F}\tilde{\boldsymbol{z}}(\omega_m) \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{x}}^T(\omega_1) \\ \tilde{\boldsymbol{x}}^T(\omega_2) \\ \cdots \\ \cdots \\ \tilde{\boldsymbol{x}}^T(\omega_m) \end{bmatrix}\boldsymbol{\Theta} \quad (29)$$

When a complex error vector $\boldsymbol{\varepsilon}$ is introduced, which accounts for noise and nonlinearities, the preceding equations can be rewritten in the general form $\boldsymbol{Y} = X\boldsymbol{\Theta} + \boldsymbol{\varepsilon}$ with conventional definitions for $\boldsymbol{Y}$, X, and $\boldsymbol{\Theta}$. Thus, the problem can be formulated as an LS regression problem with the following complex cost function:

$$J = \tfrac{1}{2}(\boldsymbol{Y} - X\boldsymbol{\Theta})^*(\boldsymbol{Y} - X\boldsymbol{\Theta}) \quad (30)$$

The solution is given by

$$\hat{\boldsymbol{\Theta}} = [\mathrm{Re}(X^*X)]^{-1}\mathrm{Re}(X^*\boldsymbol{Y}) \quad (31)$$

where the asterisk indicates a complex conjugate transpose. Note that the cost function is made of a summation over m frequencies of interest. In addition, the covariance matrix of the estimates of $\hat{\boldsymbol{\Theta}}$ is computed as

$$\mathrm{cov}(\hat{\boldsymbol{\Theta}}) = \boldsymbol{E}\{(\hat{\Theta} - \Theta)(\hat{\Theta} - \Theta)^*\} = \sigma^2(\hat{\boldsymbol{\Theta}}) \cdot [\mathrm{Re}(X^*X)]^{-1} \quad (32)$$

where $\sigma^2(\hat{\boldsymbol{\Theta}})$ is the equation error variance and can be estimated online using

$$\hat{\sigma}^2(\hat{\Theta}) = [1/(m-p)][(\boldsymbol{Y} - X\hat{\boldsymbol{\Theta}})^*(\boldsymbol{Y} - X\hat{\boldsymbol{\Theta}})] \quad (33)$$

where p is the number of parameters to be estimated and m is the number of frequency points. Furthermore, the standard deviation of the estimation error for the lth unknown of the p parameters in $\hat{\boldsymbol{\Theta}}$ can be evaluated as the square root of the (l, l) coefficient (main-diagonal coefficient) of the covariance matrix. This standard deviation allows for an online assessment of the accuracy of the parameter estimates.

The type of required online calculations should also be analyzed for an assessment of the computational effort for real-time applications. For a given frequency ω_n, the DTFT at the ith time step is related to the DTFT at the $(i-1)$th time step as follows:

$$\tilde{x}_i(\omega_n) = \tilde{x}_{i-1}(\omega_n) + x_i\exp(-j\omega_n i\Delta t) \quad (34)$$

An important feature of this technique is that the time-domain data from previous flight maneuvers, containing good information for PID purposes, can still be used by simply iterating the calculation of the DTFT. Thus, the DTFT approach allows for retaining all of the PID results from previous time steps and, at the same time, provides the necessary flexibility to follow changes in the system dynamics.

In terms of frequency range, the m frequencies over which the cost function is evaluated can be selected as evenly spaced between $\omega_{\min}$ and $\omega_{\max}$. Typically, the rigid-body dynamics frequency range for the considered aircraft can be selected allowing filtering out higher frequency noise and/or structural interference. Clearly, a smaller range of frequency would decrease the computational effort.

Time-Domain-Based Method for Online PID

Among time-domain-based PID techniques, the LS algorithm is the most widely used approach. The reliability of this method comes from the property that a pseudoinverse solution for a set of overdetermined linear equations is optimal in the LS sense.

For the purpose of online estimation within real-time applications this method is usually modified into the RLS algorithm to get the information from the data being updated. It is known that an adaptive algorithm with information-dependent data forgetting can also be added.

However, the RLS method has not always shown consistent accuracy for aircraft PID purposes. One problem is the lack of robustness of the recursively computed pseudoinverse.[11] An additional problem

is the lack of a reliable computation of the estimation error variance to be used within a logic to decide on the accuracy of the estimates.

An alternative method in the time domain is the locally weighted regression (LWR) PID algorithm.[22–24] This method is essentially an LS method modified with local weighting and the retention scheme of most valuable data to the current flight point. First, let us consider an ordinary linear regression problem,

$$\boldsymbol{Y} = X\boldsymbol{\Theta} + \varepsilon \tag{35}$$

where $\boldsymbol{Y} \in R^n$ is the measurement vector, $X \in R^{n \times p}$ is the matrix of independent variables that denote the given flight point, $\boldsymbol{\Theta} \in R^p$ is the parameter vector to be estimated, and ε is the measurement error vector with zero mean and $\text{cov}\{\varepsilon\} = \sigma^2 I$.

For aircraft PID purposes, there are six equations in the body axes related to the six aerodynamic coefficients C_X, C_Y, C_Z, C_l, C_m, and $C_{n.}$. Consider, for example, the coefficient C_m: At a given time step, C_m can be evaluated with the available measurements of motion and control variables. When the process for n data points is repeated, the measurement vector $\boldsymbol{Y} \in R^n$, which is a vector of n values of C_m, is formulated. Each row of X, with size p, is given by the measurements of the motion and control variables ($1, \alpha, q, \delta_{S\_R}$, and δ_{S_L}) at each data point. Note that a constant 1 is included in each vector to estimate C_{m_0}. When the process for n data points is iterated, the complete matrix X is evaluated. Finally, the vector with the unknown parameters to be estimated, $\boldsymbol{\Theta}$, contains the aerodynamic stability and control derivatives, that is, (c_{m_0}, c_{m_α}, c_{m_q}, $c_{m_{\delta S\_L}}$, and $c_{m_{\delta S\_R}}$). The solution to the problem is given in the LS sense[10] by

$$\hat{\boldsymbol{\Theta}} = (X^T X)^{-1} X^T \boldsymbol{Y} \tag{36}$$

To obtain estimates that are focused on the current flight point within the flight envelope, a weighting matrix can be introduced in the Eq. (36) so that more importance can be placed on the equations containing data points closer to the current point. For this purpose, the elements of the diagonal weighting matrix are given by

$$\boldsymbol{W}_{ii} = \exp\left(-d_i \big/ 2k^2\right) \tag{37}$$

where d_i is the norm of the difference between the current flight point and the ith one and k is a time-varying Gaussian window width that increases as the variance increases, so that W becomes closer to the unity matrix considering a greater number of points. Similarly, k decreases when the variances decrease, so that W becomes more selective, placing more weight on the data points that are closer to the current point. With the multiplication by the weighting matrix, the problem is redefined as

$$W\boldsymbol{Y} = WX\boldsymbol{\Theta} + W\varepsilon \tag{38}$$

The goal is to find the value of $\boldsymbol{\Theta}$ minimizing the following weighted sum of square errors:

$$Q = \varepsilon^T W^2 \varepsilon = (W\boldsymbol{Y} - WX\boldsymbol{\Theta})^T (W\boldsymbol{Y} - WX\boldsymbol{\Theta}) \tag{39}$$

The solution to this problem is given by

$$\hat{\boldsymbol{\Theta}} = (WX)^+ W\boldsymbol{Y} = (X^T W^2 X)^{-1} X^T W^2 \boldsymbol{Y} \tag{40}$$

It can be shown that this solution is such that the weighted error vector

$$\boldsymbol{We} = W\boldsymbol{Y} - WX\hat{\boldsymbol{\Theta}} \tag{41}$$

is unbiased. The covariance matrix of the solution is given by

$$\begin{aligned}\text{cov}\{\hat{\Theta}\} &= \boldsymbol{E}\{(\hat{\Theta} - E\{\hat{\Theta}\})(\hat{\Theta} - E\{\hat{\Theta}\})^T\} \\ &= (X^T W^2 X)^{-1} X^T W^2 \cdot \text{cov}\{Y\} \cdot W^2 X (X^T W^2 X)^{-1} \\ &= \sigma^2 (X^T W^2 X)^{-1} X^T W^4 X (X^T W^2 X)^{-1}\end{aligned} \tag{42}$$

Furthermore, it can be shown that an optimal estimation for σ^2 is the mean square error (MSE) value:

$$\text{MSE} = \frac{(\boldsymbol{Y} - X\hat{\boldsymbol{\Theta}})^T (\boldsymbol{Y} - X\hat{\boldsymbol{\Theta}})}{n - p} \tag{43}$$

Therefore, the standard deviation of estimation error of $\hat{\boldsymbol{\Theta}}$ can be calculated online as the square root of the covariance matrix $\text{cov}\{\hat{\Theta}\}$ using the MSE in lieu of σ^2.

Typically, online estimation problems impose constraints on the size of the data matrix $D = WX$. Therefore, once the number of rows in D reaches its maximum user-defined value, the problem arises of which rows in D should be replaced by the new data. A simple strategy would be to delete the oldest rows, or the ones that are in temporal space farthest from the current one. However, this strategy could cause the deletion of rows that bring precious information to the estimation, inducing, therefore, ill conditioning and an increase in the estimation variances.

An enhanced approach used in this study is to delete the rows of D that brings less information for the estimation process, so that when those rows will be replaced, it is likely that the new rows will bring more information than the ones just deleted. This would then cause a decrease for the variances of the estimates. This goal is achieved by replacing the rows of D whose deletion causes the trace of $(D^T D)^{-1}$ to increase the least [because the less is this trace, the more $(D^T D)^{-1}$ is well conditioned]. In turn, this decreases the variances of the estimated parameters. For that purpose, let us denote D as follows:

$$D = \begin{bmatrix} Z \\ H \end{bmatrix} \tag{44}$$

Then

$$(D^T D)^{-1} = \left(\begin{bmatrix} Z^T & H^T \end{bmatrix} \begin{bmatrix} Z \\ H \end{bmatrix} \right)^{-1} = (Z^T Z + H^T H)^{-1} \tag{45}$$

Denoting $P \equiv Z^T Z + H^T H$, $\quad R \equiv Z^T Z$, and applying the matrix inversion lemma,

$$R^{-1} = (P - H^T H)^{-1} = P^{-1} - P^{-1} H^T (HP^{-1} H^T - I)^{-1} HP^{-1} \tag{46}$$

Thus,

$$\text{tr}(R^{-1}) = \text{tr}(P^{-1}) - \text{tr}[P^{-1} H^T (HP^{-1} H^T - I)^{-1} HP^{-1}] \tag{47}$$

Because $HP^{-1}H^T$ is a scalar,

$$\begin{aligned}&\text{tr}[P^{-1} H^T (HP^{-1} H^T - I)^{-1} HP^{-1}] \\ &\quad = \frac{\text{tr}(P^{-1} H^T HP^{-1})}{HP^{-1} H^T - I} = \frac{HVS^{-2} V^T H^T}{HVS^{-2} V^T H^T - I}\end{aligned} \tag{48}$$

where VSV^T is the singular value decomposition (SVD) of P. The problem is then reduced to find the particular row H among all of the rows of D maximizing the earlier trace. Because HV can be any row of the matrix DV, let $C = HV$. Then,

$$\begin{aligned}&\text{tr}[P^{-1} H^T (HP^{-1} H^T - I)^{-1} HP^{-1}] \\ &\quad = \sum_{j=1}^{p} \frac{c_{ij}^2}{s_{jj}^2} \Bigg/ \sum_{j=1}^{p} \frac{c_{ij}^2}{s_{jj}} - 1 \equiv F(i)\end{aligned} \tag{49}$$

Thus, $k = \max_I [F(i)]$ is the index of the row to be deleted. In practice, because often only some of the p parameters are actually needed, the partial trace of $(D^T D)^{-1}$ can also be used [which is the sum of the diagonal elements of $(X^T X)^{-1}$ corresponding to the important parameters].

Comparison of the Different PID Results

The primary objective of this study was to compare the interface of the described FTR and LWR PID schemes with the aerodynamic modeling of the postfailure conditions using the performance metrics outlined for real-time applications.

The comparison was conducted through three distinct studies. For each of these studies the availability of a failure detection and identification (FDI) scheme was assumed. Furthermore, it was assumed, without any loss of generality, that the FDI scheme requires a 1.5-s computational time to identify clearly the failed/damaged

Table 2 Sensor noise standard deviations

Parameter	Value
Variances for measurement noise from different sensors	
Airspeed indicator	10 ft/s
Roll rate gyro	0.15 deg/s
Pitch rate gyro	0.15 deg/s
Yaw rate gyro	0.15 deg/s
Normal accelerometer	1.0 ft/s^2
Lateral accelerometer	1.0 ft/s^2
Longitudinal accelerometer	1.0 ft/s^2
Attitude roll gyro	0.1 deg
Attitude pitch gyro	0.1 deg
Attitude yaw gyro	0.1 deg
Angle of attack	0.15 deg
Sideslip angle	0.15 deg
Dryden turbulence model	
$L_u = L_v = L_w$	580 m = 1903 ft
$\sigma_u = \sigma_v = \sigma_w$	3m/s = 9.84ft/s

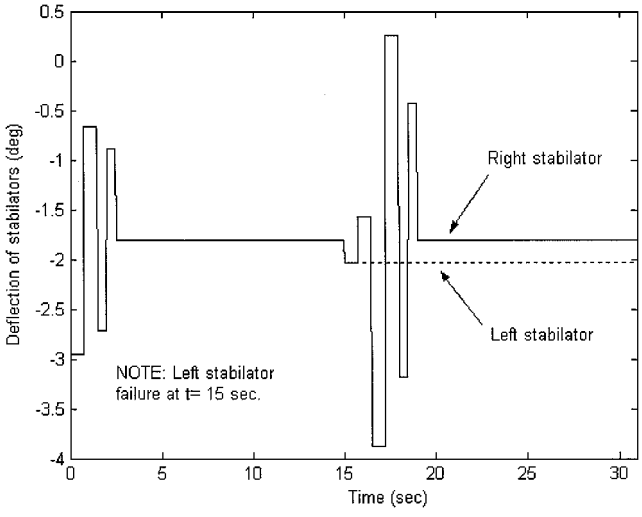

Fig. 2 Study 1, time histories of longitudinal control surface deflections.

control surface. The other assumption was the availability of the angular deflection data for all control surfaces. During the simulations, the presence of noise was modeled by adding Gaussian noise with zero mean and known standard deviations on the generic jth measurable output, except control surface deflections. The presence of system noise was also modeled using a Dryden turbulence model (see Ref. 16). Noise levels of 0, 100, and 200% were considered for the simulations, where 100% implies 1 σ of noise level. The statistical parameters of the measurement and system noise are shown in Table 2.

In the first study, the application of the scheme to a failure simulation was conducted under desirable conditions for PID purposes with different noise levels. A longitudinal simulation with online PID was started at nominal conditions with a doublet, a typical PID maneuver. After a few seconds during which the estimates converged to steady-state values, a failure involving a stuck left elevator at a midrange position with a 67% effectiveness reduction was simulated along with a second PID maneuver for the remaining healthy right stabilator. The idea was to obtain benchmark results for the online PID scheme in the best possible scenario for PID purposes.

The results of this study are shown in Figs. 2–6 for the case with a 1-σ noise level on the flight data. Figure 2 shows the stabilator maneuver with the same deflections for both sides until the failure/damage occurrence followed by a stuck left stabilator. The direct results of the online PID algorithm are shown in Figs. 3–6. The failure isolation logic provides the PID algorithm with the information that the left and right stabilators have to be considered separate outputs. With the use of information, the PID algorithm modifies the relative mathematical model by adding an additional control input; note that the PID algorithm at nominal conditions could not proceed using separate left and right stabilators because at nominal conditions these two control surfaces are completely correlated.

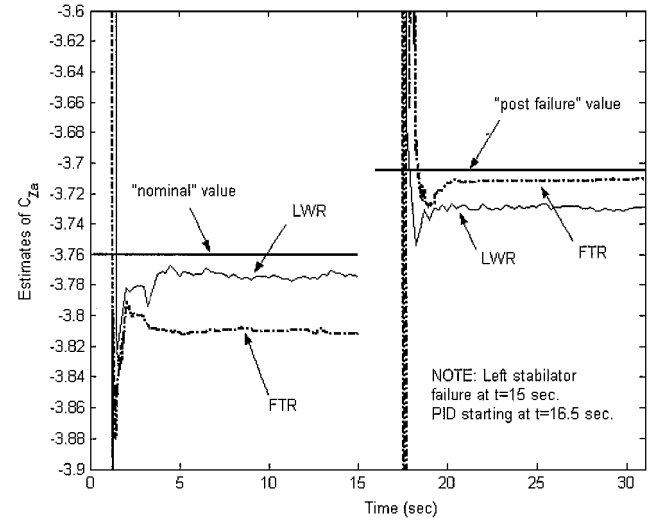

Fig. 3 Study 1, time histories of the estimates of c_{z_α}.

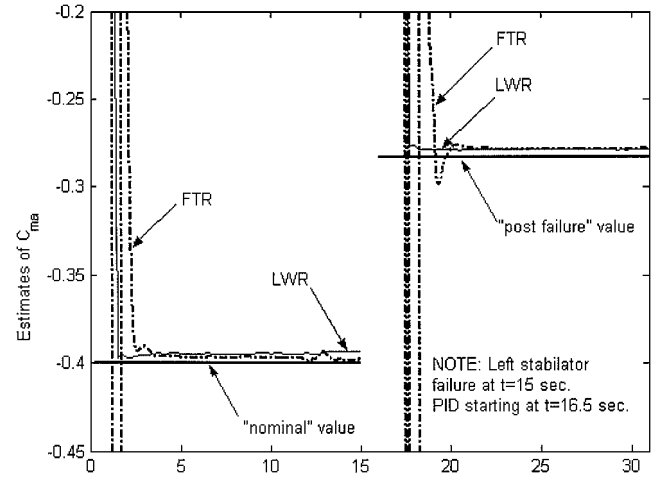

Fig. 4 Study 1, time histories of the estimates of c_{m_α}.

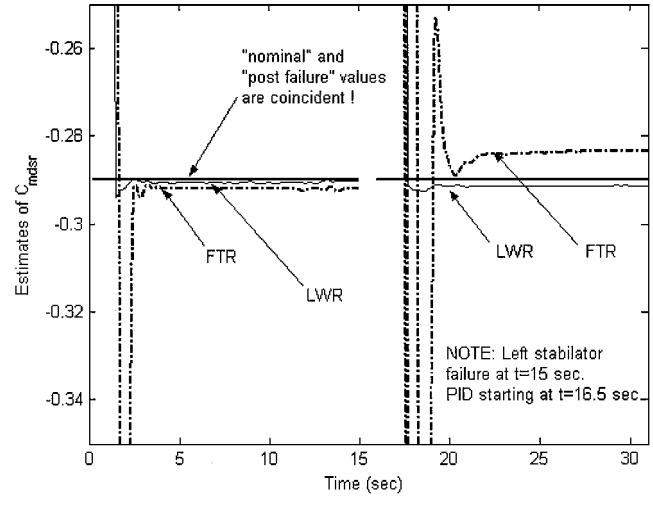

Fig. 5 Study 1, time histories of the estimates of $c_{m_{\delta S_{-R}}}$.

Figure 3 shows the online estimates of the parameter c_{Z_α} using the LWR and the FTR methods. Figure 3 highlights the failure/damage effects showing a slight decrease for c_{Z_α} at postfailure conditions. Both PID schemes show a desirable convergence of the estimates at both nominal and postfailure conditions.

Table 3 Study 1, statistics of PID estimates at nominal conditions for different noise levels

Parameter	True value	0% Noise level[a] FTR estimate σ_{EE}	0% Noise level[a] LWR estimate σ_{EE}	100% Noise level[a] FTR estimate (mean, standard deviation) average σ_{EE}	100% Noise level[a] LWR estimate (mean, standard deviation) average σ_{EE}	200% Noise level[a] FTR estimate (mean, standard deviation) average σ_{EE}	200% Noise level[a] LWR estimate (mean, standard deviation) average σ_{EE}
C_{Z_α}	−3.759	−3.8015 (0.0020)	−3.7928 (0.0008)	(−3.7992, 0.0419) (0.0412)	(−3.7489, 0.0679) (0.0228)	(−3.7824, 0.0712) (0.0699)	(−3.6114, 0.1169) (0.0422)
$C_{Z_q} + C_{Z_{\dot\alpha}}$	−2.657	−2.7284 (0.0693)	−2.6793 (0.0190)	(−2.7368, 1.2712) (0.9240)	(−2.9975, 1.5498) (0.7619)	(−3.0684, 2.7667) (1.6868)	(−4.2202, 2.6892) (1.6301)
$C_{Z_{\delta s\_R}}$	−0.2	−0.1972 (0.0014)	−0.1997 (0.0005)	(−0.1961, 0.0258) (0.0285)	(−0.2060, 0.0445) (0.0159)	(−0.2051, 0.0561) (0.0539)	(−0.2710, 0.0854) (0.0341)
C_{m_α}	−0.4	−0.3907 (0.0033)	−0.3948 (9.0e−7)	(−0.3962, 0.0038) (0.0025)	(−0.3916, 0.0059) (0.0043)	(−0.3958, 0.0069) (0.0053)	(−0.3771, 0.0108) (0.0048)
$C_{m_q} + C_{m_{\dot\alpha}}$	−4.0	−4.6145 (0.1133)	−4.0155 (2.0e−5)	(−4.5483, 0.1197) (0.0557)	(−4.0387, 0.1241) (0.1446)	(−4.5700, 0.2752) (0.1277)	(−4.1275, 0.2527) (0.1854)
$C_{m_{\delta s\_R}}$	−0.29	−0.2941 (0.0024)	−0.2906 (5.2e−7)	(−0.2945, 0.0027) (0.0017)	(−0.2917, 0.0043) (0.0030)	(−0.2954, 0.0058) (0.0041)	(−0.2979, 0.0081) (0.0039)
MATLAB[b] CPU time, s				2.74	20.21	2.74	20.21
MATLAB[b] FLOPS				33,623,835	185,458,861	33,623,835	185,458,861

[a]Note σ_{EE} indicates standard deviation of the estimation error. [b] For a 31-s simulation.

Table 4 Study 1, statistics of PID estimates at postfailure conditions for different noise levels[a]

Parameter	True value	0% Noise level[b] FTR estimate σ_{EE}	0% Noise level[b] LWR estimate σ_{EE}	100% Noise level[b] FTR estimate (mean, standard deviation) average σ_{EE}	100% Noise level[b] LWR estimate (mean, standard deviation) average σ_{EE}	200% Noise level[b] FTR estimate (mean, standard deviation) average σ_{EE}	200% Noise level[b] LWR estimate (mean, standard deviation) average σ_{EE}
C_{Z_α}	−3.705	−3.6858 (0.0072)	−3.7386 (0.0005)	(−3.6812, 0.0313) (0.0223)	(−3.7205, 0.0445) (0.0229)	(−3.6574, 0.0645) (0.0556)	(−3.6394, 0.1018) (0.0475)
$C_{Z_q} + C_{Z_{\dot\alpha}}$	−2.04	−2.2781 (0.3125)	−2.0696 (0.0181)	(−2.3536, 1.3689) (0.8491)	(−2.8921, 1.8209) (0.9920)	(−2.9040, 2.7200) (2.0561)	(−4.6059, 3.2661) (2.0208)
$C_{Z_{\delta s\_R}}$	−0.2	−0.1984 (0.0066)	−0.1991 (0.0004)	(−0.1999, 0.0311) (0.0201)	(−0.1956, 0.0400) (0.0211)	(−0.2262, 0.0640) (0.0472)	(−0.1957, 0.0809) (0.0435)
C_{m_α}	−0.283	−0.2763 (0.0055)	−0.2792 (1.53e−6)	(−0.2758, 0.0021) (0.0012)	(−0.2778, 0.0029) (0.0060)	(−0.2743, 0.0044) (0.0025)	(−0.2717, 0.0065) (0.0061)
$C_{m_q} + C_{m_{\dot\alpha}}$	−2.595	−2.9231 (0.2411)	−2.6022 (0.0001)	(−2.9325, 0.0952) (0.0474)	(−2.6589, 0.1034) (0.2592)	(−2.9611, 0.1898) (0.0911)	(−2.7911, 0.2231) (0.2647)
$C_{m_{\delta s\_R}}$	−0.29	−0.2832 (0.0051)	−0.2904 (1.14e−6)	(−0.2835, 0.0025) (0.0011)	(−0.2901, 0.0027) (0.0055)	(−0.2846, 0.0046) (0.0021)	(−0.2900, 0.0047) (0.0056)
$C_{l_{\delta s\_R}}$	−0.0046	−0.0048 (0.0001)	−0.0046 (8.8e−6)	(−0.0047, 0.0007) (0.0002)	(−0.0043, 0.0012) (0.0003)	(−0.0045, 0.0012) (0.0003)	(−0.0045, 0.0021) (0.0005)

[a]MATLAB CPU time and MATLAB FLOPS same as in Table 3. [b]Note σ_{EE} indicates standard deviation of the estimation error.

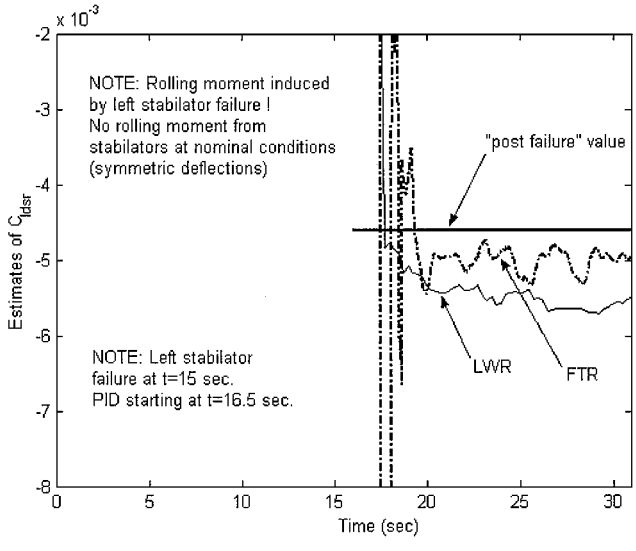

Fig. 6 Study 1, time histories of the estimates of $c_{l_{\delta S\_R}}$.

A similar trend for the results is shown in Fig. 4 relative to the parameter c_{m_α}. Note that c_{m_α} experiences a substantial decrease at postfailure conditions. In terms of estimation of the control derivatives, the most important results are shown in Figs. 5 and 6 relative to the parameters $c_{m_{\delta S\_R}}$ and $c_{l_{\delta S\_R}}$ at postfailure conditions. Note that both parameters are not introduced at nominal conditions, that is, when $c_{m_{\delta S\_R}} = c_{m_{\delta S\_L}} = 1/2 c_{m_{\delta S}}$ and $c_{l_{\delta S\_R}} = -c_{l_{\delta S\_L}}$.

The results for both PID methods for the first study at different noise levels are summarized in Tables 3 and 4 relative to nominal and postfailure conditions, respectively, in terms of the value of the estimates (after convergence) as well as the value of the standard deviation of the estimation error. Note that the statistics in Tables 3 and 4 were evaluated using data from 100 runs for each of the studies.

An important result of this study is the comparison of the computational effort required by the two PID algorithms shown in Table 3. In particular, the FTR PID method uses approximately 13.5 and 18% of the MATLAB® CPU time and MATLAB floating point operations (FLOPS), respectively, required by the LWR PID method. To understand the reasons for such a drastic difference in the computational efforts, the nature of the computations for the two methods must be evaluated. The computations in the FTR PID start at each step with the Fourier transformations applied to the dynamic equations

Table 5 Study 2, statistics of PID estimates at postfailure conditions for different noise levels[a]

Parameter	True value	0% Noise level[b] FTR estimate σ_{EE}	0% Noise level[b] LWR estimate σ_{EE}	100% Noise level[b] FTR estimate (mean, standard deviation) average σ_{EE}	100% Noise level[b] LWR estimate (mean, standard deviation) average σ_{EE}	200% Noise level[b] FTR estimate (mean, standard deviation) average σ_{EE}	200% Noise level[b] LWR estimate (mean, standard deviation) average σ_{EE}
C_{Z_α}	−3.705	−3.7023 (0.0417)	−3.7498 (0.0009)	(−3.5898, 0.1425) (0.8402)	(−3.3586, 0.4557) (0.6299)	(−3.4761, 0.2915) (0.7769)	(−2.7256, 0.7076) (0.9392)
$C_{Z_q} + C_{Z_{\dot\alpha}}$	−2.04	1.3936 (1.9436)	−1.6245 (0.0519)	(−2.0465, 6.7399) (23.9420)	(1.0261, 10.301) (29.058)	(−1.5937, 13.412) (22.5705)	(3.5567, 13.936) (44.7511)
C_{m_α}	−0.283	−0.2472 (0.0172)	−0.2792 (2.18e−6)	(−0.2675, 0.0133) (0.0252)	(−0.2673, 0.0190) (0.0491)	(−0.2689, 0.0223) (0.0144)	(−0.2087, 0.0267) (0.0779)
$C_{m_q} + C_{m_{\dot\alpha}}$	−2.595	−2.7405 (0.8004)	−2.6050 (0.0001)	(−2.5102, 0.5949) (0.7186)	(−1.7945, 0.3928) (2.2651)	(−2.2039, 1.0089) (0.4181)	(−0.8891, 0.4078) (3.7107)

[a]MATLAB CPU time and MATLAB FLOPS same as in Table 3. [b]Note σ_{EE} indicates standard deviation of the estimation error.

of motion, which are linear algebraic equations with constant unknown parameters. Next the FTR algorithm proceeds to perform the linear regression leading to the inversion of the matrix $\mathrm{Re}(X^TX)$. In this effort the inversion is performed using the SVD method. Particularly, for each vector Θ of parameters to be estimated, one SVD [$\mathcal{O}(n^3)$ FLOPS] of the matrix $\mathrm{Re}(X^TX)$ (average size 6×6) is performed at each step.

The nature of the computations in the LWR PID is quite different and substantially more complicated. In fact, in the LWR algorithm, the linear regression is applied directly to the linearized dynamic equations in the time domain. Therefore, to set up these equations, a block of data has to be prepared at each step. In this study, 50 data points at every 10 time steps are used. Furthermore, to evaluate the online estimate of the variance of the estimation error, a retention and deletion is performed at each step while arranging the regression equations. Finally, at every 10 time steps, the newly formed linear regression equation is solved using the calculated local weighting matrix. The same SVD used for the FTR algorithm is used in the LWR algorithm for the inversion in the linear regression equations (size 50×50) and for the retention and deletion scheme (average size 6×6). The size of these two computations, in addition to the computations for the weighting matrix, is the main reason for the enormous computational effort associated with the LWR method. Clearly the amount of computations will depend on the number of data points and the updating frequency set by the user. To decrease the computational effort, a smaller size of block data or a lower frequency for the update can be used. In this study, the selection of 50 data points at every 10 sampling time step has provided the best compromise between computational efforts and accuracy of the results.

In the second study, the application of the scheme to a failure simulation was conducted under the worst possible conditions for PID purposes, that is, assuming that no significant excitation of the aircraft dynamics, other than the excitation due to the failure/damage itself, is available for PID purposes. In other words, no control surface deflections on the remaining healthy control surfaces are commanded for the purpose of aiding the postfailure PID process.

As expected, both PID schemes experience difficulties in converging to accurate estimates in the absence of PID excitation, as shown in Figs. 7 and 8 for the estimates of c_{Z_α} and c_{m_α}, respectively. Note that the FTR and LWR estimates for $c_{m_{\delta S\_R}}$ and $c_{l_{\delta S\_R}}$ are not available because the right stabilator is never deflected in this study. A complete summary of the statistical results for different levels of noise is provided in Table 5.

From the results of the second study, it is clear that some form of excitation could enhance the PID performance at postfailure/damage conditions. This important issue has been discussed with some details in Ref. 25. In fact, at postfailure conditions, the deflections of multiple control surfaces within some form of control allocation scheme for an overactuated aircraft can substantially increase the complexity of the PID problem. Two PID stages were formulated as a solution to this problem.[25,26] The first phase is for the estimates of the main derivatives without a specific PID maneuver, whereas the second phase is for the estimates of the control derivatives of

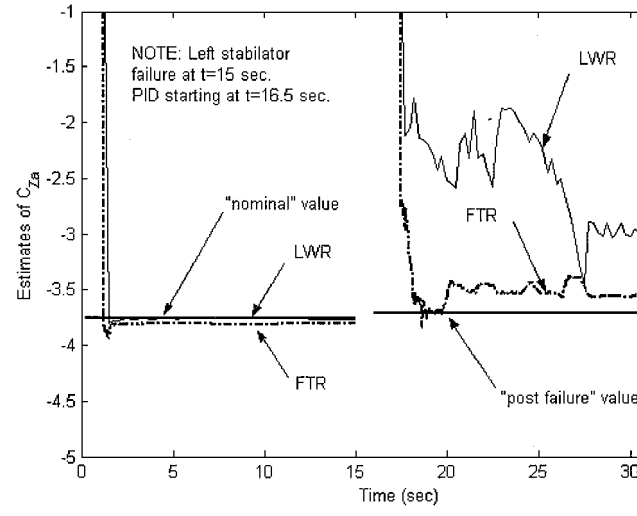

Fig. 7 Study 2, time histories of the estimates of C_{z_α}.

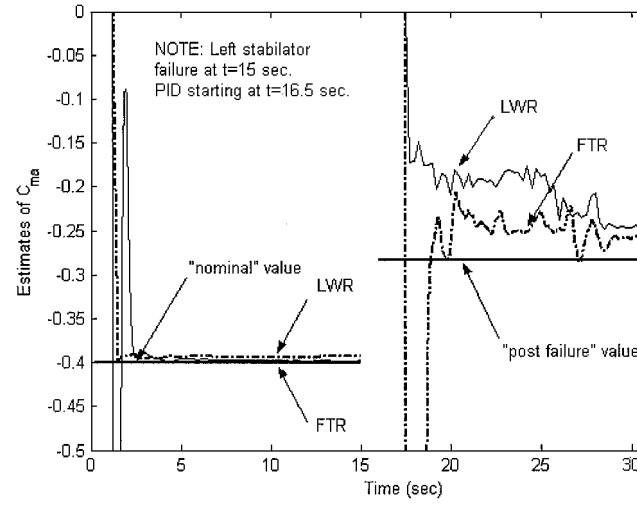

Fig. 8 Study 2, time histories of the estimates of C_{m_α}.

the individual surfaces, deployed for accommodation purposes, to be used in the online control allocation algorithm.

The solution proposed in this paper consists of activating a short preprogrammed specific PID maneuver following a positive FDI. This is very important because initiating a PID process following a failure/damage is potentially very critical. Nevertheless, because the postfailure/damage handling qualities may be already seriously compromised, it can be acceptable to take on an additional small hazard to allow a successful PID following some form of PID excitation so that accurate real-time estimates could ultimately be provided to the control laws.

Table 6 Study 3, statistics of PID estimates at postfailure conditions for different noise levels[a]

Parameter	True value	0% Noise level[b] FTR estimate σ_{EE}	0% Noise level[b] LWR estimate σ_{EE}	100% Noise level[b] FTR estimate (mean, standard deviation) average σ_{EE}	100% Noise level[b] LWR estimate (mean, standard deviation) average σ_{EE}	200% Noise level[b] FTR estimate (mean, standard deviation) average σ_{EE}	200% Noise level[b] LWR estimate (mean, standard deviation) average σ_{EE}
C_{Z_α}	−3.705	−3.7032 (0.0070)	−3.7415 (0.0002)	(−3.6547, 0.0722) (0.0830)	(−3.6119, 0.1440) (0.0672)	(−3.6799, 0.1355) (0.1742)	(−3.6780, 0.2347) (0.1468)
$C_{Z_q} + C_{Z_{\dot\alpha}}$	−2.04	−2.0961 (0.3103)	−2.0680 (0.0063)	(−2.8371, 3.0353) (3.0454)	(−4.8894, 3.9351) (2.9737)	(−3.6002, 5.2200) (6.3154)	(−4.9589, 7.1155) (6.6061)
$C_{Z_{\delta s\_R}}$	−0.2	−0.1913 (0.0060)	−0.1989 (0.0001)	(−0.2225, 0.0730) (0.0624)	(−0.1983, 0.0871) (0.0579)	(−0.1942, 0.1019) (0.1251)	(−0.1679, 0.1630) (0.1260)
C_{m_α}	−0.283	−0.2741 (0.0045)	−0.2792 (3.01e−7)	(−0.2732, 0.0051) (0.0046)	(−0.2704, 0.0097) (0.0062)	(−0.2734, 0.0081) (0.0080)	(−0.2745, 0.0078) (0.0112)
$C_{m_q} + C_{m_{\dot\alpha}}$	−2.595	−3.1545 (0.1988)	−2.6022 (1.1e−5)	(−3.0995, 0.2216) (0.1684)	(−2.7906, 0.2538) (0.2764)	(−3.0582, 0.4747) (0.2893)	(−2.4497, 0.2096) (0.5049)
$C_{m_{\delta s\_R}}$	−0.29	−0.2900 (0.0038)	−0.2904 (2.15e−7)	(−0.2898, 0.0061) (0.0034)	(−0.2896, 0.0050) (0.0054)	(−0.2889, 0.0114) (0.0057)	(−0.2893, 0.0055) (0.0096)
$C_{l_{\delta s\_R}}$	−0.0046	−0.0053 (0.0003)	−0.0046 (1.84e−6)	(−0.0051, 0.0019) (0.0003)	(−0.0050, 0.0030) (0.0007)	(−0.0052, 0.0019) (0.0008)	(−0.0045, 0.0025) (0.0011)

[a]MATLAB CPU time and MATLAB FLOPS same as in Table 3. [b]Note σ_{EE} indicates standard deviation of the estimation error.

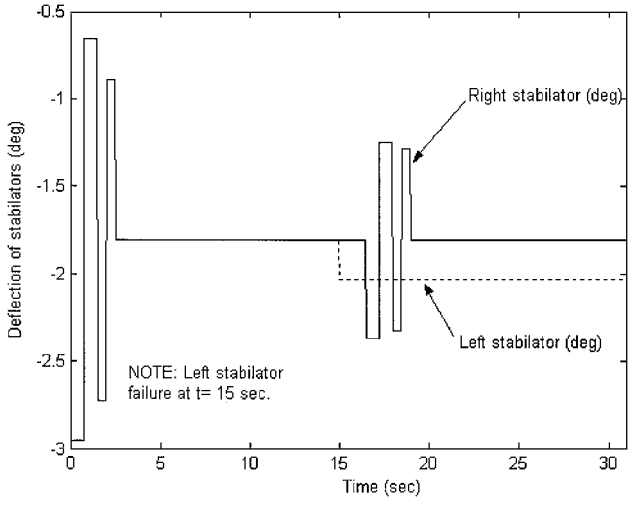

Fig. 9 Study 3, time histories of longitudinal control surface deflections.

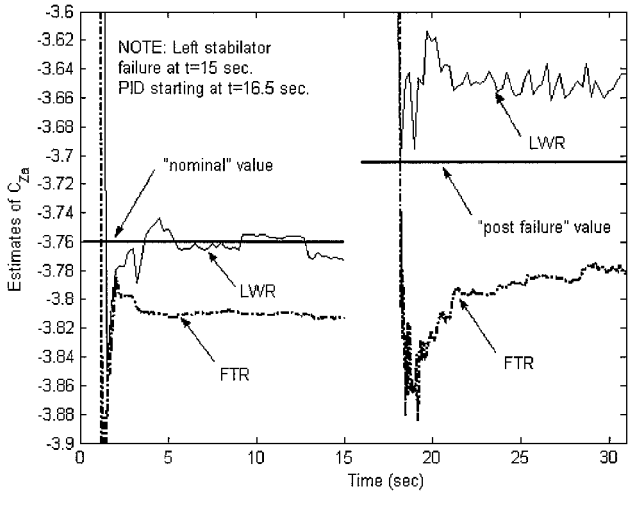

Fig. 10 Study 3, time histories of the estimates of c_{z_α}.

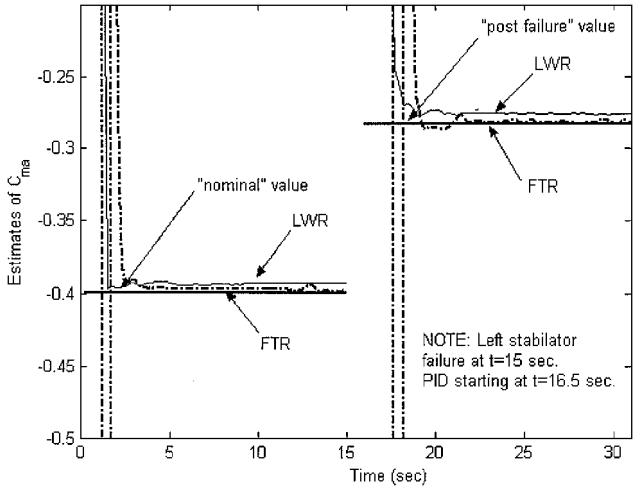

Fig. 11 Study 3, time histories of the estimates of c_{m_α}.

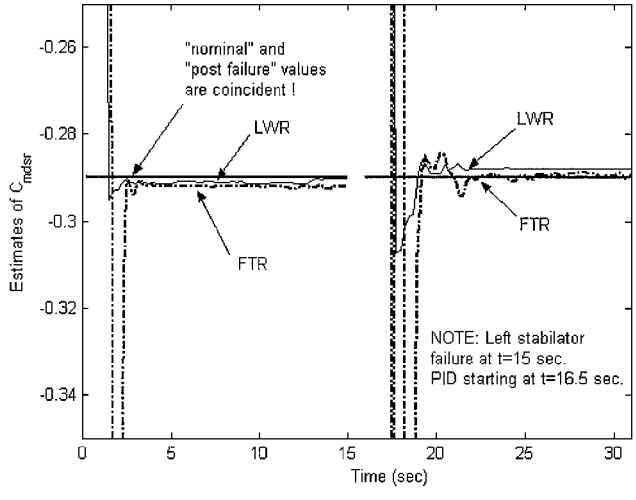

Fig. 12 Study 3, time histories of the estimates of $c_{m_{\delta S\_R}}$.

Thus, a third scenario is proposed, where a short PID maneuver is simulated, following a positive FDI, activating the remaining healthy stabilator. The only requirements for this preprogrammed PID maneuver is to be brief and to induce just enough excitation for PID purposes without further endangering the safety of the aircraft. Within this effort no specific studies for the selection of this maneuver were conducted; in fact, the only condition was for this PID maneuver to induce a Δg response in the range of $\pm 0.5\ g$. The results of this study are summarized in Figs. 9–13. Figure 9 shows the deflections of the both sides of the stabilators with the failure on the left side and the postfailure PID maneuver for the right side. The improvements in the results, with respect to the preceding study, for the estimates of c_{Z_α}, c_{m_α}, $c_{m_{\delta S\_R}}$, and $c_{m_{\delta S\_L}}$ are shown in Figs. 10–13 for both PID methods. A complete summary of the statistical results for different levels of noise is provided in Table 6. The clear conclusion for this study is that a short and small-magnitude postfailure

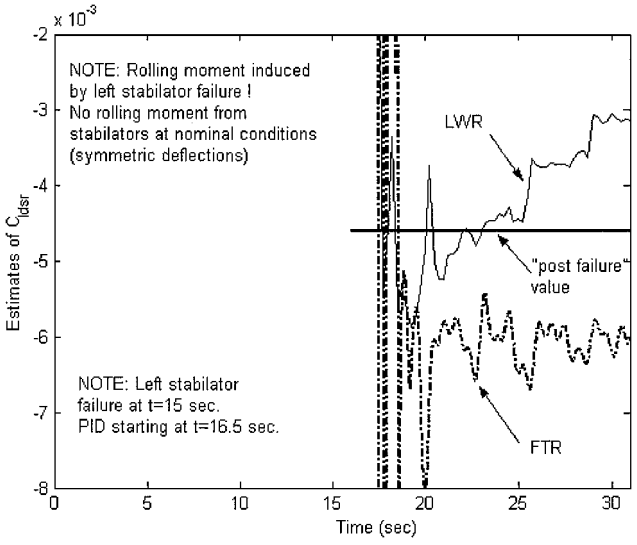

Fig. 13 Study 3, time histories of the estimates of $c_{l_{\delta S\_R}}$.

PID phase can be very useful in allowing the online PID scheme to provide accurate and reliable estimates of all of the primary stability and control derivatives at postfailure conditions.

Conclusions

The study in this paper has focused on a comparison of the performance of two recent PID techniques to provide online real-time estimates of the aircraft stability and control derivatives following failure/damage to a primary control surface. The first technique, based on the time domain, was implemented as presented in the literature. The second technique, based in the frequency domain, was substantially modified from its original formulation to allow direct estimates of the aerodynamic derivatives. The study has also emphasized the advantages of using ad hoc small and short preprogrammed PID maneuvers to help the PID process following the failure/damage occurrence.

The comparison of the performance of the PID schemes has been conducted through a simulation study featuring a detailed modeling of the postfailure conditions. Both methods feature the availability of an online quantification of the statistical reliability of the estimates.

Although conceptually different, the FTR and the LWR techniques present similar and desirable performance for studies 1 and 3. As expected, the performances of both schemes deteriorate without an appropriate level of excitation (study 2). Furthermore, both schemes exhibit desirable robustness as the noise level increases. For the LWR method, this robustness is due to the retention and deletion scheme along with the weighting mechanism. The FTR's robustness is due to that the algorithm operates within a limited frequency range and, thus, it naturally filters noise at frequencies outside the range.

Whereas the two methods have very similar performance in terms of accuracy of the estimates, convergence time, and robustness to noise, the FTR method clearly outperforms the LWR method in terms of required computational effort.

Acknowledgments

Support for the first author was provided by a 2000 Korea Research Foundation Grant. Partial support for the second, third, and fourth authors has been provided through a Grant from the Institute for Software Research, Fairmont, WV.

References

[1]Iliff, K. W., and Taylor, L. W., "Determination of Stability Derivatives from Flight Data Using a Newton–Raphson Minimization Technique," NASA TN D-6579, 1972.

[2]Iliff, K. W., and Maine, R. E., "Practical Aspects of Using a Maximum Likelihood Estimation Method to Extract Stability and Control Derivatives from Flight Data," NASA TN D-8209, 1976.

[3]Maine, R. E., and Iliff, K. W., "Application of Parameter Estimation to Aircraft Stability and Control the Output-Error Approach," NASA Reference Publ. 1168, June 1986.

[4]Napolitano, M. R., Naylor, S., Neppach, C., and Casdorph, V., "Online Learning Nonlinear Direct Neurocontrollers for Restructurable Control Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 170–176.

[5]Monaco, J., Ward, D., Barron, R., and Bird, R., "Implementation and Flight Test Assessment of an Adaptive, Reconfigurable Flight Control System," AIAA Paper 97-3738, Aug. 1997.

[6]Brinker, J. S., and Wise, K. A., "Nonlinear Simulation Analysis of a Tailless Advanced Fighter Aircraft Reconfigurable Flight Control Law," AIAA Paper 99-4040, Aug. 1999.

[7]Calise, A. J., Lee, S., and Sharma, M., "Direct Adpative Reconfigurable Control of a Tailless Fighter Aircraft," AIAA Paper 98-4108, Aug. 1998.

[8]Totah, J., "Adaptive Flight Control and Online Learning," AIAA Paper 97-3537, Aug. 1997.

[9]"Intelligent Flight Control: Advanced Concept Program—Final Report," The Boeing Co., Rept. Boeing-STL 99P0040, May 1999.

[10]Mendel, J. M., *Discrete Techniques of Parameter Estimation,* Marcel Dekker, New York, 1973.

[11]Ljung, L., *System Identification: Theory for the User,* Prentice–Hall, Englewood Cliffs, NJ, 1987.

[12]Bodson, M., "An Information-Dependent Data Forgetting Adaptive Algorithm," Proceedings of the 1995 American Control Conf., June 1995.

[13]Smith, L., Chandler, P. R., and Patcher, M., "Regularization Techniques for Real-Time Identification of Aircraft Parameters," AIAA Paper 97-3740, Aug. 1997.

[14]Chandler, P., Patcher, M., and Mears, M., "System Identification for Adaptive and Reconfigurable Control," *Journal of Guidance, Control, and Dynamics,* Vol. 18, No. 3, 1995, pp. 516–524.

[15]Gelb, A., *Applied Optimal Estimation,* MIT Press, Cambridge, MA, 1973.

[16]Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls: Part I and II*, Roskam Aviation and Engineering Corp., Ottawa, KS, 1994.

[17]Hoak, D. E., and Finck, R. D., "USAF Stability and Control DATCOM," U.S. Air Force Wright Aeronautical Labs., AFWAL-TR-83-3048, Wright–Patterson AFB, OH, Oct. 1960; rev. 1978.

[18]Morelli, E. A., "Real-Time Parameter Estimation in the Frequency Domain," AIAA Paper 99-4043, Aug. 1999.

[19]Morelli, E. A., "In-Flight System Identification," AIAA Paper 98-4261, Aug. 1998.

[20]Morelli, E. A., "High Accuracy Evaluation of the Finite Fourier Transform Using Sampled Data," NASA TM 110340, June 1997.

[21]Klein, V., "Aircraft Parameter Estimation in Frequency Domain," AIAA Paper 78-1344, Aug. 1978.

[22]Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W., *Applied Linear Regression Models,* 3rd ed., Richard D. Irwin, Inc., Burr Ridge, IL, 1996.

[23]Atkeson, C. G., Moore, A. W., and Schaal, S., "Locally Weighted Learning," *Artificial Intelligence Review*, Vol. 11, Feb. 1997, pp. 11–73.

[24]Atkeson, C. G., Moore, A. W., and Schaal, S., "Locally Weighted Learning for Control," *Artificial Intelligence Review*, Vol. 11, Feb. 1997, pp.75–113.

[25]Patcher, M., Chandler, P. R., Nelson, E., and Rasmussen, S., "Parameter Estimation for Overactuated Aircraft with Nonlinear Aerodynamics," AIAA Paper 99-4046, Aug. 1999.

[26]Buffington, J., Chandler, P. R., and Patcher, M., "Integration of Online System Identification and Optimization-Based Control Allocation," AIAA Paper 98-4487, Aug. 1998.