On-line parameter estimation for indirect adaptive flight control: a practical evaluation of several techniques

Georges Hardier, Gilles Ferreres, Masayuki Sato

On-line parameter estimation for indirect adaptive control: a practical comparison of several techniques

G. Hardier, G. Ferreres, and M. Sato

Abstract— **To be able to maintain their performance, the Flight Control Systems in use on most transport aircraft should be endowed with resilient properties to deal with sensor/actuator faulty conditions or any significant airframe changes. During the H2020 VISION project, several adaptive control designs have been developed by ONERA, and were flight tested to this end using the MuPAL-α research aircraft of JAXA. One of these solutions involve a gain-scheduled controller designed off-line plus an estimation algorithm to track the time-varying parameters on-line, permitting the values of the main stability and control derivatives required to schedule the controller to be updated in real time. This paper focuses on the estimation part, and provides an experimental comparison of some results achieved in strictly similar conditions by using either frequency or time domain techniques. A new Hybrid Stabilized Recursive Least-Squares (HSRLS) is also proposed, as well as competing options regarding the aircraft modeling used by the estimator.**

I. INTRODUCTION

Fly-By-Wire (FBW) systems have paved the way for continuous improvements in the reliability and performance of the Flight Control Systems (FCS) used by most civil and military aircraft. But these advanced capabilities increase the complexity of the FCS which, in turn, require many data sources to be available, such as the Angle of Attack (AoA) and the Calibrated Air Speed (CAS). Embedding resilience in the system thus becomes essential to deal with faulty situations of the corresponding sensors, in order to avoid switching to degraded control modes (i.e. to reduced control laws or flight envelope protections) [7]. When these key parameters are no more available or reliable enough, it is still possible to schedule the control laws according to estimated parameters instead of measured ones to keep the same level of performances. Two possible alternatives are: 1/ to replace the faulty measurement by an estimated value (*virtual sensing*) and to keep going with a scheduling strategy using this estimate [9]; 2/ to schedule the laws with another type of parameters, namely some aerodynamic coefficients [6]. This paper is related to a design belonging to the second category. The underlying FDD/FTC (Fault Detection & Diagnosis/Fault Tolerant Control) scheme which has been developed is based on indirect adaptive control; the principle is to identify some of the model parameters (FDD) and to adjust a controller on-line in terms of the updated estimates (FTC).

This work took place as part of the recent H2020 European/Japanese project VISION (Validation of Integrated Safety-enhanced Intelligent flight cONtrol). The objective of this project (2016-2019) was to validate smarter technologies for aircraft Guidance, Navigation and Control (GNC) by developing vision-based systems (out of the scope of this paper, see [23]) and advanced detection and resilient methods (w3.onera.fr/h2020_vision). Today, the advanced FDD/FTC methods developed by the academic community are not fully accepted by the aerospace industry yet, resulting in a gap between the current know-how and the industrial practice implemented aboard the planes [24]. Several reasons explain this gap: the low Technology Readiness Level (TRL) of the academic works, real-time implementation issues, and the certification constraints. That is why past projects jointly led by industry and European academic stakeholders (e.g. projects ADDSAFE or RECONFIGURE) have involved high fidelity simulators and implementation constraints to address realistic issues [7]. On the other hand, the VISION project intended to increase the TRL of the designs by implementing and testing them on experimental platforms [23].

During VISION, ONERA has developed and flight tested an indirect adaptive control strategy, which can be seen as an integrated FDD/FTC approach [12]. A gain-scheduled flight control law is designed off-line at first, by using a state-space modal technique to synthesize a controller which affinely depends only on the major model aerodynamic parameters [5]. Then, a real time estimation algorithm is implemented to track these time-varying parameters on-line, and to schedule the controller w.r.t. the provided estimates [11], i.e. the updated values of the model stability and control derivatives (aerodynamic parameters). Considering their relevance for industry, the scenarios evaluated during VISION regarding this adaptive strategy include a loss of airspeed information (sensor fault), a loss of efficiency (actuator fault), or a variation or misknowledge of some aerodynamic coefficients (due to external phenomena like icing or a loss of reference condition following a sensor fault). However, the identification scheme could be used by itself for FDD purposes only, by monitoring the changes in the control or stability derivatives to diagnose some unexpected behaviour [8].

In 2017-2018, a number of ground and flight tests were achieved to implement and to evaluate this adaptive control strategy aboard the MuPAL-α aircraft of JAXA. During these works, a Frequency Domain (FD) technique was used to perform the estimation task (see [6,11] for a description of the techniques and [12] for results). In 2019, additional Time Domain (TD) estimation methods were also developed and implemented: 1/ to be compared to the FD approach in the same experimental conditions since they are much simpler and less demanding; 2/ because such techniques were required for estimating the controller gains of a direct adaptive approach (also tested during VISION). This work reports some results of the ground tests achieved to complete this practical comparison (unfortunately, flight testing could no longer be performed in 2019 due to technical issues of the aircraft).

G. Hardier and G. Ferreres are with ONERA/DTIS (The French Aerospace Lab, Department of Information Processing and Systems), Toulouse, France, [(+33)5 62 25 27 77; Georges.Hardier@onera.fr].

M. Sato is with Japan Aerospace Exploration Agency, 6-13-1 Osawa, Mitaka, Tokyo, Japan.

The paper is organized as follows. The principles of the TD estimation techniques are given in §II, including the proposed new HSRLS algorithm. Two possible formulations of the aircraft model are also described in §III, and a comparison of the test results achieved thanks to the experimental platform of JAXA is presented in §IV.

II. TIME DOMAIN ESTIMATION ALGORITHMS

A. *The basics in a nutshell*

On-line identification makes up a special class of model-based FDD methods as the residuals refer to parameters instead of system variables. Estimating the model parameters in real time faces several problems: lack of measurement redundancy, external disturbances (turbulence), and poor data content (e.g., using only the control signals available about a steady flight condition). The challenge is to design a technique that can be quite insensitive to the disturbances but can detect unexpected events within a short delay. Contrary to the batch case, the use of enriched excitation inputs is prohibited, and advanced methods must be developed to deal with possibly ill-conditioned problems by including strong forms of regularization in the process. Both TD and FD techniques can be considered for such purposes. Whatever it takes place in the FD or in the TD, estimation can be based on recursive algorithms (making use of continuous measurements as soon as they are available), or on sequential procedures (processing moving data windows with a lower sampling rate) to produce a succession of piecewise constant estimates. In this paper, we will consider only recursive algorithms, and compare the FD implementation of [12] with some of the TD algorithms.

Looking for the simplest approaches w.r.t. implementation issues, we will assume in the sequel of §II that Linear-in-their-Parameters (LP) models can be used to represent, at least locally, the system dynamics. This assumption permits on-line methods based on Equation Error minimization to be developed, that usually involve only a few parameters in practical applications. The LP will be written in the form:

$$\hat{y}_k = f(x_k) = \sum_{j=1}^{m} \hat{\theta}_j \, r_j(x_k) = R_k \hat{\Theta} \qquad (1)$$

where x_k represents the independent variables available at sample k (e.g. the state and control derivatives), and the measurement y_k is assumed to be scalar in the sequel to alleviate the developments (e.g. the system output or the state derivative to be modeled). The functions $r_j(x)$ represent the regressors, while θ_j (Θ vector) are the regression parameters to be determined during the estimation process. Herein, we also suppose that the number m of regressors is defined a priori. On the other hand, we do not make any assumption about the regressors r_j, which can represent either directly the explanatory variables or some nonlinear functions (e.g., the monomials of a polynomial expansion). From k data samples (x_k, y_k) available at time t_k, the minimization of the estimation errors yields to consider a criterion of the type:

$$C(k) = \sum_{i=1}^{k} \lambda^{k-i}(y_i - f(x_i))^2 \quad [+C_{Reg}] \qquad (2)$$

where adding an [optional] regularization term permits for instance large variations of the θ_j parameters to be penalized, and thus to improve the conditioning of the problem. In the weighted version (2) of Least-Squares (LS) [19], called Exponential Forgetting (EF), a forgetting factor

$0 < \lambda \leq 1$ is introduced, resulting in decaying weights for the oldest data. Contrary to the standard LS where the algorithm attaches less and less importance to new samples, and hence is not suited to the estimation of time-varying parameters. EF favors fast adaptations with time by discounting the oldest measurements, thereby allowing parameter to be changed based on the most recent information. Assuming $C_{Reg} = 0$, a recursive formulation is preferred for on-line implementation, and yields estimation updates as soon as a new data sample (x_{k+1}, y_{k+1}) becomes available following:

$$\begin{cases} \hat{\Theta}(k+1) = \hat{\Theta}(k) + K(k+1)[y_{k+1} - R_{k+1}\hat{\Theta}(k)] \\ K(k+1) = P(k+1) \, R_{k+1}^T \end{cases} \qquad (3)$$

where the covariance matrix (inverse of the information matrix $P(k) = H^{-1}(k)$) is computed by applying the matrix inversion lemma (hence avoiding a matrix inversion):

$$P(k+1) = \left[I_m - \frac{P(k) R_{k+1}^T R_{k+1}}{R_{k+1} P(k) R_{k+1}^T + \lambda} \right] \frac{P(k)}{\lambda} \qquad (4)$$

For initializing the procedure in the absence of prior knowledge, $\Theta(0)$ can be set to 0 and the (diagonal) variances of P to large values to enforce a fast adaptation. Otherwise, assuming that the modeling error ε has zero mean value and a constant diagonal covariance $\sigma^2 I$ (homoscedasticity), the covariance of the estimation errors can be computed as $\text{Cov}[\hat{\Theta}(k)] = \sigma^2 H^{-1}(k) = \sigma^2 P(k)$. As σ^2 is usually unknown, it must be estimated from the estimation errors:

$$\hat{\sigma}^2 \approx \sum_{i=1}^{k} [y_i - R_i \hat{\Theta}(i)]^2 \Big/ (k - m) \qquad (5)$$

Choosing $\lambda \approx 0$ will allow rapid changes of the parameters to be tracked, since only recent samples will contribute to the estimates, but will also result in increasing their sensitivity to the noise. On the opposite, choosing $\lambda \approx 1$ will provide a slower adaptation and a better robustness to noise. The selection of the right value of λ must be a compromise between the adaptation speed and the accuracy of the estimates. Usual values are chosen in the range [0.9,1[and kept constant during the process, but algorithms involving a controlled variation of λ with time were also developed.

Actually, when $\lambda < 1$, it is easy to infer from (4) that $P(k)$ can become very large in case of non persistent excitation with $P(k+1) = P(k)/\lambda > P(k)$. This phenomenon, known as covariance wind-up or blow-up, results in a great sensitivity to small perturbations in the regressors that can cause large variations of the estimated parameters [15]. The challenge that improved algorithms try to take up is therefore to get some guarantees that P is kept bounded as $\kappa_1 I \leq P(k) \leq \kappa_2 I$, the 1st part of the inequality ensuring that the algorithm can track time-varying parameters reasonably well (since the gain vector $K(k)$ will remain away from 0), while the 2nd condition permits the stability of the algorithm to be guaranteed [4,19].

Consequently, a great deal of proposals for new refined algorithms has been done during the last 30 years, that are related either to the original EF version or to similar KF formulations of the algorithm [10]. The recursion often focuses on the information matrix instead of the covariance matrix since the regularization term generally prevents from the derivation of a simple formulation by using the matrix inversion lemma (as it is the case when this term is omitted).

Hence, most advanced techniques destroy the *beauty* of the Recursive LS (RLS) algorithm, that avoids a matrix inversion of order m (or having to solve a linear system of equations) to compute the gain vector $K(k)$. To comply with the stringent implementation constraints on aircraft computers, such heavy computations are prohibited beyond $m > 3$ and recursive factorization algorithms are required to enable on-line implementation. Fortunately, such algorithms do exist; let's quote Potter's Square Root [15], Bierman's U-D [1], or AUDI [17] algorithms. Taking the implementation aspects into account whenever it can lead to promising variants, the most famous (amongst the number of RLS variants proposed since the beginning of the 80s) are briefly described in [10].

Apart from implementation constraints, the other basic point to be considered is the sheer necessity of adding regularization in the estimation process, notably because all the parameters involved in (1) will not be identifiable from poor control signals. Actually, two conflicting issues arise for parameter estimation, that are difficult to reconcile. The first one is related to time-varying parameters which requires a quick adaptation property, and hence a short memory. The second one results from data collinearities induced either by steady (trimmed) flight conditions and hence constant states and controls, or by closed-loop behavior with actuator deflections computed as (linear) functions of the states, or by the coupling of some surfaces to enforce aerodynamic effects by symmetrical or opposite deviations. These collinearities do not ensure persistent excitation, even with a low level of noise, and hence require a longer memory to avoid ill-conditioning and parameter divergence. To overcome this dilemma and to prevent those numerical problems, modified versions of the RLS were early developed [2,14]. These rely on the introduction of a regularization term C_{Reg} in the cost (2). Two examples of such an approach are the *Modified Sequential Least Squares* (MSLS) [22] and the SRLS algorithm summarized in §II.B.

B. Stabilized Recursive Least Squares (SRLS)

This algorithm, developed by Bodson [3], relies on an approximated update of the dispersion matrix $H^{-1}(k)$. SRLS aims at improving the stability of the estimates (considering the presence of noise and other modeling errors), by limiting the deviations of the estimates from their previous values:

$$C(k) = \sum_{i=1}^{k} \lambda^{k-i}[y_i - R_i \Theta(k)]^2 + \eta \| \Theta(k) - \Theta(k-1) \|^2 \quad (6)$$

This is a peculiar case of the MSLS algorithm, and the batch-type updating law thus becomes:

$$\hat{\Theta}(k) = \left[\sum_{i=1}^{k} \lambda^{k-i} R_i^T R_i + \eta I \right]^{-1} \left[\sum_{i=1}^{k} \lambda^{k-i} R_i^T y_i + \eta \hat{\Theta}(k-1) \right] (7)$$

which can be expressed as: $\hat{\Theta}_k = \hat{\Theta}(k) = H^{-1}(k)G(k)$. By denoting $P_k = P(k) = H^{-1}(k)$ (for consistency with §II.A) the SRLS image of the covariance matrix, the recursive expression involved by $H(k)$ updating is [3]:

$$P_k^{-1} = \lambda P_{k-1}^{-1} + R_k^T R_k + \eta(1-\lambda)I \quad (8)$$

As opposed to a standard weighted LS for which a straightforward use of the matrix inversion lemma allows a recursive update of P_k to be derived in terms of P_{k-1}, the presence of the regularization term in (6) prevents from using the same technique. To circumvent this problem [21] suggest to replace the global regularization process (all parameter variations are penalized at the same time) by an approximately equivalent one for which the parameters are alternately considered one by one. After a cycle comprising m successive iterations, every parameter has been considered and their variations constrained, and a new cycle can start again. A sequence of vectors $u(n) = [0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0]^T$ is thus defined accordingly, where only the nth is not 0. This trick permits the matrix inversion lemma to be applied by using an intermediate matrix $M = [R_k^T \ \sqrt{m\eta(1-\lambda)} \, u(n)]$ of size $m \times 2$, leading to the updating formula [3]:

$$P_k = [P_{k-1} - P_{k-1}M \ [M^T P_{k-1}M + \lambda I_2]^{-1} M^T P_{k-1}]/\lambda \quad (9)$$

which just involves to invert a matrix of size 2×2 (so-called improved version of the SRLS algorithm). Otherwise, it can also be shown that the parameter updates obey:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + P_k R_k \, \tilde{\varepsilon}_k + \lambda \eta P_k [\hat{\Theta}_{k-1} - \hat{\Theta}_{k-2}] \quad (10)$$

where $\tilde{\varepsilon}_k = y_k - R_k \hat{\Theta}_{k-1}$. The major advantage of the SRLS algorithm lies in the fact that both P and P^{-1} are bounded [2]. Unlike RLS, P do not tend towards zero when there is continuous excitation (stopping the adaptation) and, unlike weighted LS, P do not tend towards infinity in the absence of persistent excitation (resulting in ill-conditioning and divergence). It is also noteworthy that two SLS algorithms were also proposed earlier by [14], that are more general than the SRLS of [2,3]. Finally, the asymptotic convergence of SRLS to the true parameter values can be established for the batch-type version of (7), and more recent studies [13] also advocated the advantages of SRLS in terms of transient behavior and computational issues (flops).

C. A new Hybrid Stabilized Recursive Least Squares

Unfortunately, despite its interesting properties, the SRLS algorithm suffers from some major drawbacks. Firstly, it allows only the rate of parameter variations to be weighted, but a necessary stability condition is that $P_0 \leq (1/\eta) I_m$, i.e. a trade-off between this weighting and the initial parameter uncertainty should be defined that prevents from setting $P_0 \to \infty$ and $\eta \to \infty$ at the same time (it would be nice to strongly weight parameter variations while keeping a large initial uncertainty). On the other hand, it does not allow the departure from initial values to be weighted (a priori knowledge) as achieved by standard batch LS algorithms using Bayesian regularization. Yet, benefiting from both forms of regularization is desirable and achievable, especially if we relax the requirement of avoiding a matrix inversion. To be able to combine both forms, (6) should be replaced by:

$$C(k) = \sum_{i=1}^{k} \lambda^{k-i}[y_i - R_i \Theta_k]^2 + \eta \| \Theta_k - \Theta_k^* \|^2 \quad (11)$$

where the ith component of Θ_k^* is $\Theta_{k,i}^* = \begin{cases} \Theta_{k-1,i} & \text{if } i \in I_1 \\ \Theta_{0,i} & \text{if } i \in I_2 \end{cases}$, and

I_1 and I_2 represent 2 user-defined subsets of $[1,m]$. Setting $\partial C(k)/\partial \Theta(k) = 0$ leads to:

$$\left[\sum_{i=1}^{k} \lambda^{k-i} R_i^T R_i + \eta I \right] \hat{\Theta}_k = \sum_{i=1}^{k} \lambda^{k-i} R_i^T y_i + \eta \Theta^*(k) \quad (12)$$

If we switch to a more general form permitting the η weights to be different from each other ($\Gamma = \text{diag}[\eta_1 \ \eta_2 \ \cdots \ \eta_m]$):

$$P_k = \left[\sum_{i=1}^{k} \lambda^{k-i} R_i^T R_i + \Gamma \right]^{-1} \quad \text{(with } P_0 = \Gamma^{-1}) \quad (13)$$

and $P_k^{-1} = \lambda P_{k-1}^{-1} + R_k^T R_k + (1-\lambda)\Gamma$. As a result:

$$P_k^{-1}\,\hat{\Theta}_k = R_k^T\, y_k + \lambda\, P_{k-1}^{-1}\,\hat{\Theta}_{k-1} + \Gamma[\Theta_k^* - \lambda\,\Theta_{k-1}^*] \quad (14)$$

Otherwise, we can infer from the above P_k^{-1} expression that:

$$\lambda\, P_k\, P_{k-1}^{-1}\,\hat{\Theta}_{k-1} = [P_k\, P_k^{-1} - P_k\, R_k^T R_k - (1-\lambda)\, P_k\,\Gamma]\,\hat{\Theta}_{k-1} \quad (15)$$

and finally, by combining (14) and (15), we obtain:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + P_k\, R_k^T\,\tilde{\varepsilon}_k + P_k\,\Gamma[\Theta_k^* - \lambda\,\Theta_{k-1}^* - (1-\lambda)\,\hat{\Theta}_{k-1}] \quad (16)$$

Alternatively, the right hand side of (12) becomes:

$$\sum_{i=1}^{k}\lambda^{k-i}R_i^T y_i + \Gamma\,\Theta_k^* = S_k + \Gamma\,\Theta_k^* \quad (17)$$

and the proposed resulting Hybrid SRLS algorithm (HSRLS) comes down to the recursive equations:

$$P_k^{-1} = \lambda\, P_{k-1}^{-1} + R_k^T R_k + (1-\lambda)\,\Gamma \quad (\text{with } P_0^{-1} = \Gamma) \quad (18)$$

followed by:

$$\hat{\Theta}_k = \hat{\Theta}_{k-1} + P_k\, R_k^T\,\tilde{\varepsilon}_k + P_k\,\Gamma[\Theta_k^* - \lambda\,\Theta_{k-1}^* - (1-\lambda)\,\hat{\Theta}_{k-1}] \quad (19)$$

The estimates can also be iteratively computed via the simpler recursion (from $S_0 = 0$):

$$S_k = \lambda\, S_{k-1} + R_k^T y_k \quad \text{and:}\quad \hat{\Theta}_k = P_k\,[S_k + \Gamma\,\Theta_k^*] \quad (20)$$

Remark 1: In the case where $\lambda = 1$ and $\Theta_k^* = \Theta_0$, (18) and (19) reduce to $P_k^{-1} = P_{k-1}^{-1} + R_k^T R_k$ and $\hat{\Theta}_k = \hat{\Theta}_{k-1} + P_k\, R_k^T\,\tilde{\varepsilon}_k$, which are the standard RLS update equations (no forgetting factor and regularization included in $P_0^{-1} = \Gamma$).

Remark 2: In addition, the trick (9) is preserved, that permits the matrix inversion lemma to be applied by using a matrix $M = [R_k^T \quad \sqrt{m\gamma_n(1-\lambda)}\, u(n)]$ of size $m \times 2$, where γ_n is the nth component of the diagonal matrix Γ. Over m successive iterations, the diagonal of P^{-1} is incremented by $m\gamma_n(1-\lambda)$, just as if $(1-\lambda)\,\Gamma$ was added at each of the m iterations.

III. AIRCRAFT MODELING EQUATIONS

A. Standard formulation of the modeling

During the VISION project, the longitudinal motion was considered, hence we focus on the aircraft lift, pitch and load factor equations that characterize the short-period dynamics. In the context of the adaptive scheme, at a given flight condition, the stability derivatives to be tracked (physical parameters of the aerodynamic model) appear as pseudo-constants in a linearized aerodynamic model including the four states α, q, θ, V and the output Nz (α and q are the AoA and the pitch rate, V is the true airspeed, δ_e is the elevator input, and Nz is the load factor output). The influence of the altitude variations (z) are neglected, as the aerodynamic coefficients only indirectly depend on z:

$$\begin{cases} \dot{\alpha} = Z_\alpha\alpha + Z_q q + Z_\delta\,\delta_e \quad [+Z_V V] \\ \dot{q} = M_\alpha\alpha + M_q q + M_\delta\,\delta_e \quad [+M_V V] \\ \dot{\theta} = q \\ \dot{V} = X_\alpha\alpha + X_q q + X_\delta\,\delta_e + X_V V \\ N_z = N_\alpha\alpha + N_q q + N_\delta\,\delta_e \quad [+N_V V] \end{cases} \quad (21)$$

The aircraft is supposed to fly about a steady flight condition, and a constant engine thrust is assumed. To feed the gain scheduled controller [6,12], only some lines of (21) need to be estimated in practice, i.e. those corresponding to the fast longitudinal states $\dot{\alpha}$, $\dot{q}$ and Nz. Hence, only (21.1), (21.2) and (21.5) are considered in the sequel. It is also worth noting that the stability derivatives relative to the airspeed

(denoted in square brackets) are weakly identifiable on-line, especially at a trimmed flight condition. However, even if these parameters should be strongly constrained w.r.t. to their initial values, they must be kept in the equations to avoid biasing the equation errors. A LP model is obtained:

$$[\dot{\alpha} \quad \dot{q} \quad Nz]^T = \Theta\,[\alpha \quad q \quad \delta_e \quad V]^T \quad (22)$$

where the Θ matrix includes the parameters to be tracked and estimated (12 at most), and the variables of the LP model (regressors) should be understood as departures from trim values. In the context of the adaptive scheme, the main goal of the estimator is to update the values of the 5 main control and stability derivatives $Z_\alpha, M_\alpha, M_q, M_\delta, N_\alpha$ [12] used for scheduling, though all the coefficients have to be estimated.

This Standard Formulation (SF) requires the time derivatives of α and q signals to be computed to get the two first outputs of equation (22). Different filters should be applied to the raw measurement data before using them to estimate the stability and control derivatives. The general idea is to implement exactly the same HP and LP filtering for all signals in order to avoid any phase lag in the filtered values used by the estimation process. The HP filter is a 1st order one (with a time constant of 1.5s), whereas the LP filters (used alone or in conjunction with a derivation operator) are 2nd order Bessel-type ones (with a cut-off frequency of 3Hz).

B. Alternative formulation of the modeling

For longitudinal motion, the Linear Time Invariant (LTI) state vector usually includes the 4 states $[u \quad w \quad q \quad \theta]$, i.e. the linearized values of the forward component of aircraft speed (body axis), the vertical component of aircraft speed (body axis), the pitch rate, and the pitch angle. The output vector also includes the load factor Nz, related to the states by:

$$\dot{w} = -g\, Nz + g\cos\theta + q\, u \quad (23)$$

Another equivalent formulation of the state vector is $[V \quad \alpha \quad q \quad \theta]$, see (21). They are linked by the relationships $u = V\cos\alpha$ and $w = V\sin\alpha$, or more simply by $\tan\alpha = w/u$, which leads to $\alpha = \text{atan}(w/u)$, and:

$$\dot{\alpha} = (u\,\dot{w} - w\,\dot{u})/(u^2 + w^2) \quad (24)$$

Assuming weak lateral motion, steady flight with almost constant speed about a trimmed flight condition (V_0, α_0), we can assume that $u \approx V, \dot{u} \approx 0$ and $u^2 + w^2 \approx V^2$. As a result, $\alpha \approx w/V$ and $\dot{\alpha} \approx \dot{w}/V$. On the other part, using real accelerations instead of load factors, (23) can be rewritten as:

$$\dot{w} \approx A_z^G + q\, V \quad (25)$$

where the vertical acceleration at CG position can be computed from the measured one at the IMU position as $A_z^G \approx A_z^{IMU} + x_{IMU}\,\dot{q}$ (x_{IMU} representing the coordinate of the IMU w.r.t. the CG position on the longitudinal axis).

If we return now to the lift and pitch equations that characterize the short-period dynamics, and include the stability derivatives to be tracked and estimated, the state equations (21.1) and (21.2) need to be considered (omitting the coefficients related to airspeed):

$$\begin{cases} \dot{\alpha} = Z_\alpha\alpha + Z_q q + Z_\delta\,\delta_e \\ \dot{q} = M_\alpha\alpha + M_q q + M_\delta\,\delta_e \end{cases} \quad (26)$$

The value of Z_q is generally close to unity, and hence we are left with the five major control and stability derivatives

$Z_\alpha, M_\alpha, M_q, M_\delta, N_\alpha$ used to schedule the controller, and that we wish to estimate on-line. By deriving these equations, and using (25), we can set the two alternative equations:

$$\begin{cases} \ddot{\alpha} = \ddot{w}/V = \dot{A}_z^G/V + \dot{q} = Z_\alpha(A_z^G/V + q) + Z_q\dot{q} + Z_\delta\dot{\delta}_e \\ \ddot{q} = M_\alpha(A_z^G/V + q) + M_q\dot{q} + M_\delta\dot{\delta}_e \end{cases} \quad (27)$$

Finally, this yields to (with $\dot{\alpha}_{eq} = (A_z^{IMU} + x_{IMU}\dot{q})/V + q$):

$$\begin{cases} \ddot{\alpha}_{eq} = Z_\alpha\dot{\alpha}_{eq} + Z_q\dot{q} + Z_\delta\dot{\delta}_e \\ \ddot{q} = M_\alpha\dot{\alpha}_{eq} + M_q\dot{q} + M_\delta\dot{\delta}_e \end{cases} \quad (28)$$

These equations mean that the standard estimation problem involving the regressors $[\alpha \quad q \quad \delta_e]$ and the two outputs $\dot{\alpha}$ and $\dot{q}$ (which are not directly measured and hence need to be estimated at first by pseudo-derivation of α and q measurements) can be replaced by an alternative one with the regressors $[\dot{\alpha}_{eq} \quad \dot{q} \quad \dot{\delta}_e]$ and the two outputs $\ddot{\alpha}_{eq}$ and $\ddot{q}$, permitting the same regression parameters $Z_\alpha, Z_\delta, M_\alpha, M_q, M_\delta$ to be estimated ($Z_q \approx 1$).

Regarding the missing N_α coefficient (among the five major derivatives), some extra computation is needed. By using (23) again, and linearized departure values from the trim condition (including a centered load factor Nz), we can establish that:

$$\dot{\alpha} \approx \dot{w}/V = q - gNz/V = -gN_\alpha\alpha/V + \cdots \quad (29)$$

As a result, the missing N_α coefficient can be inferred indirectly from the Z_α estimated value as: $N_\alpha \approx -VZ_\alpha/g$.

Of course, the main interest of this Alternative Formulation (AF) is to avoid using the AoA measurement which is only available from specific clinometric probes, hence requiring some tedious calibration to be related to inertial measurements in terms of delays for instance. On the other hand, this formulation requires the accelerometer signals, as well as the pitch rate, to be derived, which is not without any drawbacks as these signals contain higher frequencies and are more sensitive to noises in their pseudo-derivatives than their counterparts α and q measurements. As for the standard TD estimation scheme which makes use of the AoA information, the general idea is to implement exactly the same HP and LP filtering for all signals in order to avoid any phase lag in the filtered values used by the estimation process. The HP and LP filters (used alone or in conjunction with a derivation operator) are identical to those defined in §III.A.

IV. TEST RESULTS ACHIEVED ON MUPAL-α AIRCRAFT

MuPAL-α is a research airplane based on Dornier DO228-200, introduced to Japan Aerospace eXploration Agency (JAXA) in 1988 (Fig. 1), and improved with an experimental FBW system in 2000 [16,20]. The FBW computer receives information from Air Data Sensor and Inertial Navigation System, and controls a set of conventional control devices (ailerons, rudder, elevator, throttle levers) as well as additional Direct Lift Control (DLC) flaps [16]. The flexibility of the FBW system permits controllers to be changed in-flight (within the related safety regulations), and FDD/FTC designs to be easily implemented and flight tested in real conditions. Before flying, ground and safe validation tests of the designs are achieved first thanks to a simulation of the nonlinear flight dynamics using the actual onboard equipment (Fig. 1) This is referred to as Hardware-In-The-Loop Simulation (HILS).
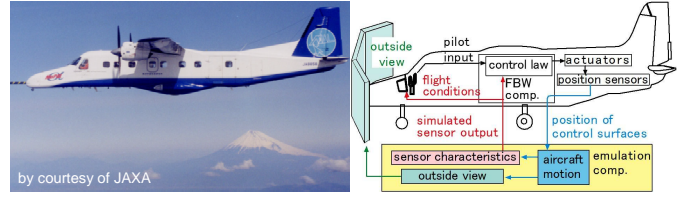

Figure 1   MuPAL-α aircraft (left) and its HILS facility (right)

To evaluate the adaptive scheme for FDD/FTC purposes, a partial Loss of Efficiency (LoE) is introduced in the elevator control channel (*control derivatives*), in addition to an initial misknowledge of the other model coefficients (*stability derivatives*) emulating some model uncertainty or a loss of reference flight point (e.g., a sensor fault leading to a lack of airspeed information). The actuator fault was introduced and parameterized in the embedded C code, to possibly modify the elevator efficiency in-flight via the control orders while being invisible from the estimator input signals [12].

Though the design of the controller is out of the scope of this paper (please report to [6,12]), before displaying test results, it is useful to stress that the architecture of the flight control law is of C*-type [18], a standard choice for longitudinal control laws (e.g., for Airbus aircraft), and that relies on load factor commands:

$$\delta_e = K_{Nz}Nz + K_q q + K_{Nzint}\int(Nz - Nz_c)dt + K_{Nzprec}Nz_c \quad (30)$$

where Nz_c is the reference acceleration input, computed via a first order filter from the stick input pilot orders. The feedforward term is typically chosen as $K_{Nzprec} = -A \times K_{Nzint}$, where the multiplicative factor A is used to tune the closed loop step responses. In addition, the time responses should remain homogeneous in the domain and some performance criteria should be met by the controller [7]: ■ dynamics should be close to a 1ˢᵗ order for tracking a reference Nz_c setting, with small overshoots in q and Nz and a given settling time; ■ no steady state error should occur in tracking the reference signal; ■ aircraft responses to external inputs should be close to the open loop behavior with improved damping.
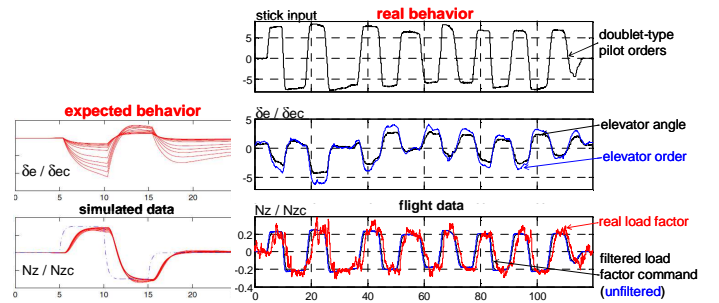

Figure 2   Performance of the controller at 120kts/2200ft in moderate turbulence and without adaptation (right) vs expected performance at the design stage for flight points between 100 and 200kts (left)

Figure 2 is a template of the plots of §IV displaying the results of the flight or HIL tests. Note that all the figures similar to this template include the following subplots: pilot stick input (upper plots), centered values of elevator order δ_{ec} in blue and real deflection δ_e in black (medium plots), load factor reference Nz_c (filtered in black, non filtered in blue) and real Nz value in red (lower plots). They also include the plots of the airspeed V_a (in black) and ground speed V_g (in red) expressed in knots. As shown by Fig. 2, for

stick input doublet signals, a satisfactory behavior of the control law is achieved, which corresponds to the expected aircraft responses at the design stage (left part of the figure).

These doublets are classically used by aircraft industry to check the performances of the control laws; they are also used by the pilots for altitude capture following changes in the flight conditions. Hence, these signals should also provide information to the estimation algorithm (with a rather poor excitation level) for tracking the changes in the model parameters. Here, a series of doublets is used in most of the tests for several purposes: 1/ to provide information to the estimation process at the beginning of the test or after the occurrence of the actuator damage 2/ to check the control performances before adaptation 3/ to display the controller performances after the adaptation is activated. The objective is not to consistently excite the system (which would not be realistic), but to be able to visualize the positive effects of the adaptation within a single test of limited duration.

In the following figures of §IV, additional plots (in the bottom part of the figures) also display the estimated values of the five main coefficients (in black solid line) used to schedule the controller [6,12], compared with those of the LTI pre-flight models (in blue dotted line). For M_δ, the red dotted line represents the LTI theoretical value including the LoE reduction, which is usually properly estimated at the end of the first doublet input following its occurrence.

The tests displayed in Fig. 3-6 take full advantage of the HILS facility by replaying the same scenarios in different mastered wind conditions. The idea is to evaluate the indirect adaptive control strategy and to compare the FD and TD estimators in more and more challenging conditions. Consequently, the results displayed are divided in two subsets of tests: (a) with no turbulence at first, (b) with strong turbulence. Large variations of the speed are also tolerated in order to check the behaviour of the process not just about a steady flight condition (e.g., 100-150kts in Fig. 3b).

Various scenarios are also simulated for this comparison, the sudden LoE occurring before or after the adaptation starts. In the latter case (Fig. 4-5), a double type of adaptation is achieved: ▪ at t=40, the performances are improved thanks to estimated values compensating for a wrong initial mismatch (emulating an error of 50 knots in the airspeed value used to schedule the initial controller); ▪ from t=70, the performances degraded by the LoE are progressively recovered as soon as new information is fed into the estimation process by excitation signals and the M_δ estimate tends to the faulty value. The 2ⁿᵈ improvement might be more progressive as the estimated M_δ parameter should converge again to the real efficiency. However, usually, M_δ converges rapidly to the faulty value (remember that the red dotted line in the bottom part of the figures represents the LTI theoretical value including the LoE reduction). The general trend is that the controller nominal performances are recovered as soon as the adaptive process is activated and fed with estimated scheduling parameters.

As regards the FD estimation (Fig. 3), we can remark that, in very unfavourable conditions (small excitation and large disturbances), the convergence of the estimates is slightly degraded and slowed down (see Fig. 3b from t=40 to t=80),
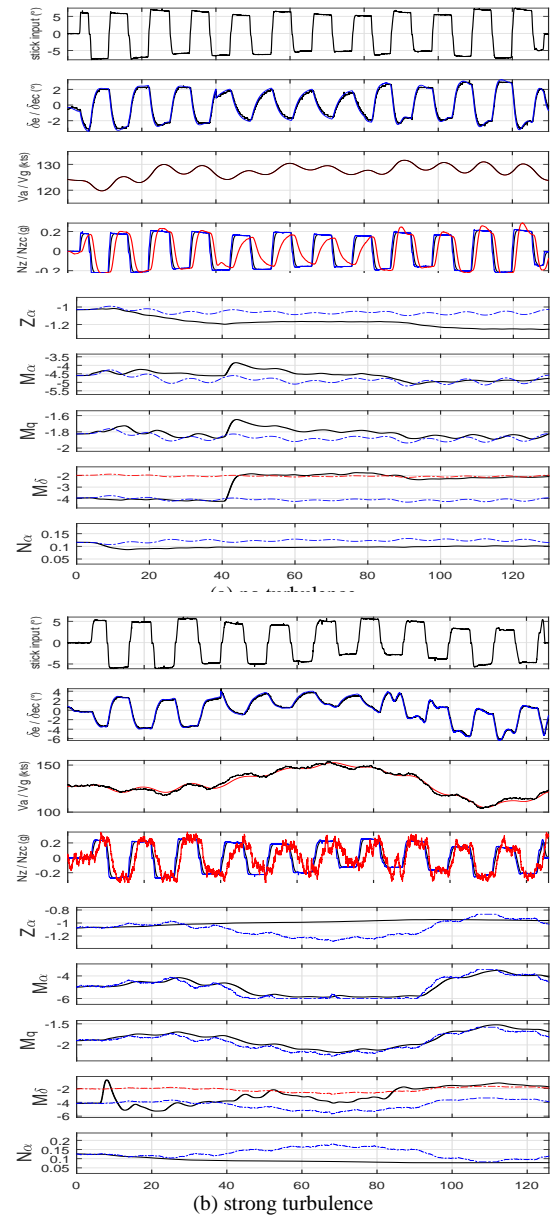

Figure 3   FD estimation (top: no turbulence, bottom: strong turbulence) with adaptation from t=80 and elevator 50% LoE at t=40

but that the parameter accuracy finally improves as soon as the amplitude of the excitation signals increases, after the adaptation is activated and proper values of the control gains tend to be recovered. It is also worth noting that the FD process acts as a memory that accumulates information and hence the convergence time depends on the value of the forgetting factor introduced in the algorithm. For these tests, a value $\lambda = 0.995$ was used, and this contributes to a slow and progressive improvement of the estimates. Apart from this transient behaviour, the performance (estimator and controller) is not impaired by the disturbances. Finally, let us remark that a slight overshoot can be observed on the load factor responses to the doublet solicitations (Fig. 3a from t=90), which does not exist in the flight tests [12]. This is explained by the variation of the actuator gain from flight to ground conditions (a ratio of about 0.7) which was not compensated in the controller gains for these HIL tests.
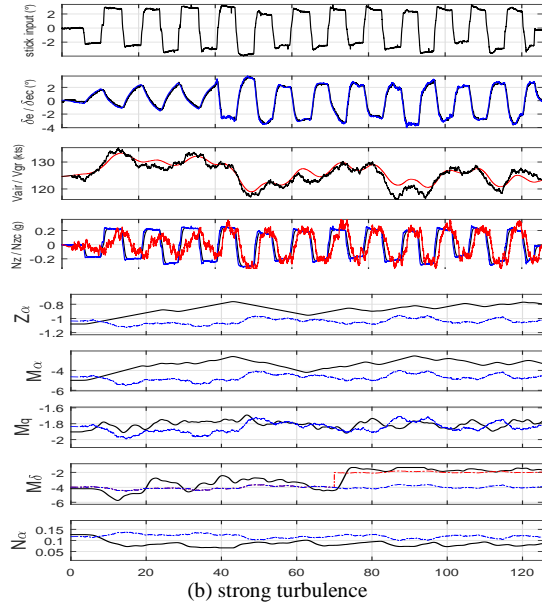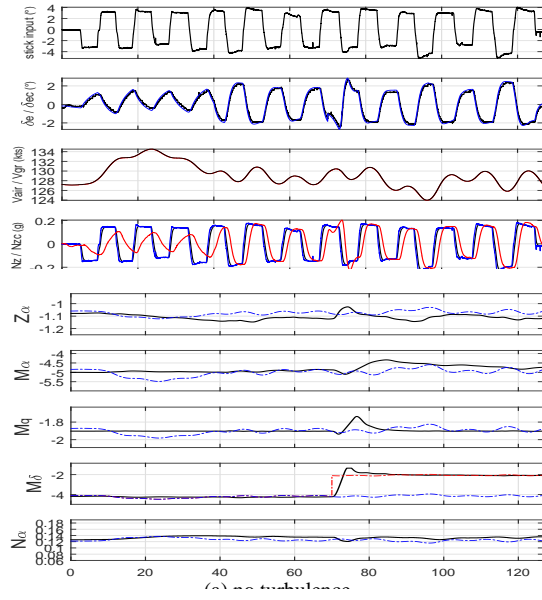
Figure 4   TD-SF estimation (top: no turbulence, bottom: strong turbulence) with adaptation from t=40 and elevator 50% LoE at t=70

Figure 5   TD-AF estimation (top: no turbulence, bottom: strong turbulence) with adaptation from t=40 and elevator 50% LoE at t=70

As regards the TD estimation (Fig. 4-5), with low to medium turbulence, both formulations give satisfactory performances in terms of control, even if some trends and forerunners can be observed for the AF when the excitation level is low. However, from only a medium level of turbulence, some estimates are strongly degraded (e.g. Z_α, M_α) in the AF case. As a result, the controller behaviour becomes oscillatory in transient stages (about t=80 on Fig. 5b), till the estimates possibly improve thanks to the excitation increase (medium turbulence not shown here). It is noteworthy that, thanks to the HSRLS algorithm, both estimators were tuned to improve their robustness to the disturbances. Hence, in case of strong turbulence, the performance of the adaptive controller remains acceptable in the SF case (Fig. 4b), but the improvement was not sufficient to get satisfactory performances in the AF case (Fig. 5b), due to large biases in some estimates.
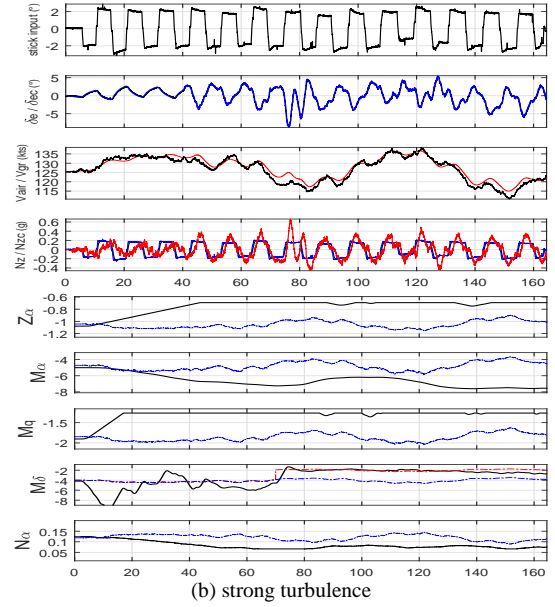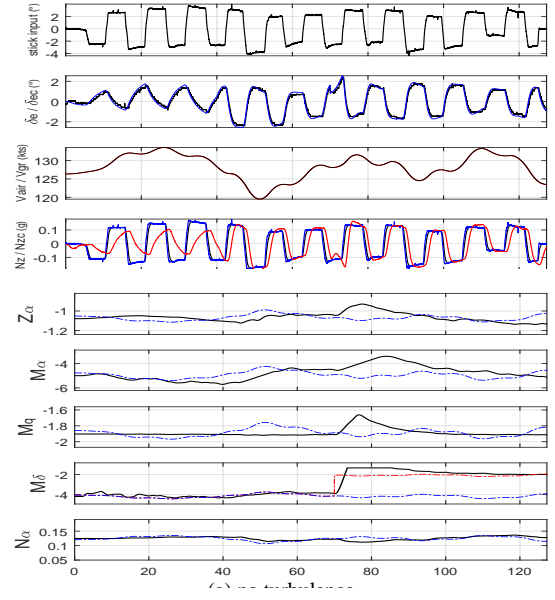
Finally, Fig. 6 shows the results achieved by simulating realistic continuous changes in altitude or in airspeed in medium turbulence conditions, to check the robustness of the adaptive process in operational scenarios. In Fig. 6a, the TD SF-based estimation is used during a scenario in which airspeed varies between 190kts and 100kts and a 50% elevator LoE occurs at t=60. After the adaptation has started (at t=30), the performances are satisfactory, the forgetting factor permitting most of the parameters to be tracked quite correctly with some delay despite their high rate of variation. In Fig. 6b, the FD estimator is used in a different scenario simulating a change of altitude (climb from 2000ft to 2700ft in the 1st half of the test, followed by a descent back to 2000ft in the 2nd half) during which a 50% elevator LoE also occurs at t=60. Despite a very low level of excitation provided by the pilot during the climb, the M_δ ends up converging to the right value.

(a) TD estimation in varying flight conditions (airspeed changes)

(b) FD estimation in varying flight conditions (altitude changes)
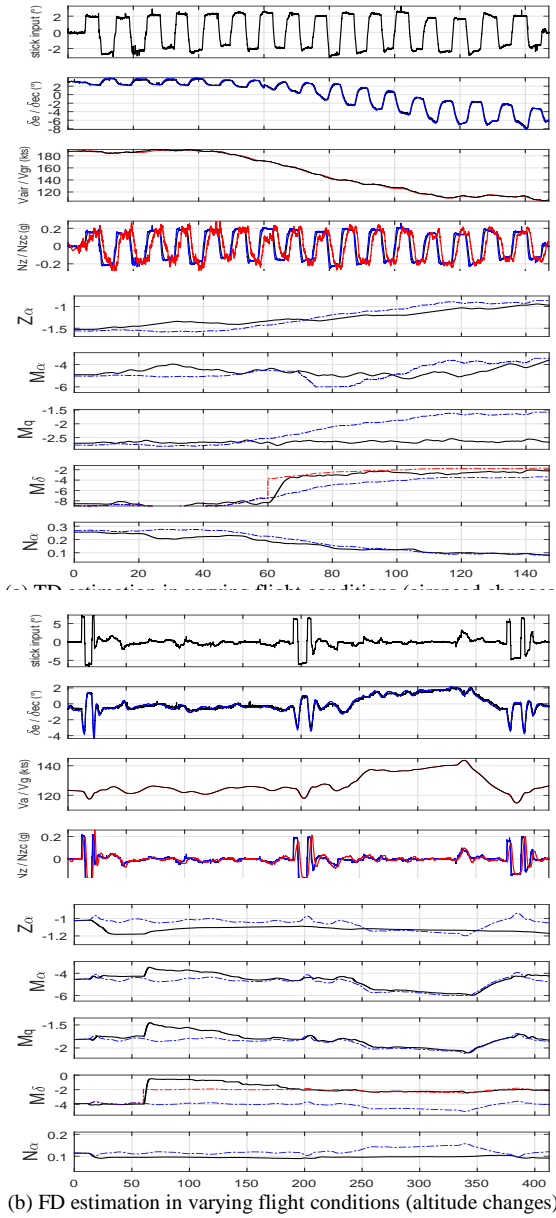
Figure 6   Tests in varying flight conditions, medium turbulence, and elevator LoE at t=60 (top: airspeed changes, bottom: altitude changes)

V. CONCLUSIONS

To sum up, the various tests displayed in §IV show that the SF-based FD and TD algorithms give satisfactory results, and much better than the TD one based on the AF. The FD estimator appears to be more robust to higher levels of disturbance, especially regarding some of the estimates, such as Z_α, M_α which rely on the derivation of the accelerometer signals in the AF case. It could also be worth testing an AF-based FD scheme, that would simplify the filtering/derivation process and could improve the accuracy thanks to an indirect filtering of the disturbances (in the FD, only a limited amount of frequencies is processed within the bandwidth of interest). Otherwise, the proposed new HSRLS algorithm was successfully implemented and allowed satisfactory results to be obtained with the SF-based TD estimator, which would not have been possible with a standard SRLS algorithm (both a priori knowledge and rate of variation need to be constrained).

REFERENCES

[1] G.J. Bierman: Factorization methods for discrete sequential estimation, Academic Press, New York, 1977.

[2] M. Bodson: An adaptive algorithm with information-dependent data forgetting, *American Control Conference*, Seattle, USA, June 1995.

[3] M. Bodson, J.E. Groszkiewicz: Multivariable adaptive algorithms for reconfigurable flight control, *IEEE Transactions on Control Systems Technology*, 1997.

[4] L. Cao, H.M. Schwartz: Analysis of the KF based estimation algorithm: an orthogonal decomposition approach, *Automatica*, 40:5–19, 2004.

[5] G. Ferreres, G. Hardier, C. Seren: Adaptive control of a civil aircraft through online parameter estimation, *Systol'16*, Barcelona, Spain, 2016.

[6] G. Ferreres, G. Hardier: Adaptive LFT control of a civil aircraft with online frequency-domain parameter estimation, *Int^al Journal of Robust and Nonlinear Control*, https://doi.org/10.1002/rnc.3993, Dec. 2017.

[7] P. Goupil *et al.*: An overview of the FP7 RECONFIGURE project: industrial, scientific and technological objectives, *Safeprocess*, Paris, 2015.

[8] G. Hardier, A. Bucharles: On-line parameter identification for in-flight aircraft monitoring, *27th ICAS Congress*, Nice, France, 2010.

[9] G. Hardier, C. Seren, P. Ezerzere: Model-based techniques for virtual sensing of longitudinal flight parameters, *International Journal of Applied Mathematics and Computer Science*, 25(1), March 2015.

[10] G. Hardier: An extended U-D algorithm with multiple forgetting factors for RLS estimation of model parameters, *Safeprocess*, Paris, 2015.

[11] G. Hardier, G. Ferreres, C. Seren: A recursive estimation algorithm to track aircraft model parameters, *Systol'16*, Barcelona, Spain, 2016.

[12] G. Hardier, G. Ferreres, M. Sato: Design and flight testing of an adaptive gain-scheduled controller using on-line model estimation, *CCTA*, Copenhagen, Denmark, Aug. 2018.

[13] C. Kamali, A.A. Pashilkar, J.R. Raol: Evaluation of RLS algorithm for parameter estimation in aircraft real time applications, *Aerospace Science and Technology*, 15:165–174, 2011.

[14] G. Kreisselmeier: Stabilized least-squares type adaptive identifiers, *IEEE Transactions on Automatic Control*, 35(3):306–310, 1990.

[15] L. Ljung, T. Soderstrom: *Theory and practice of recursive identification*, MIT Press, Cambridge, 1983.

[16] K. Masui and Y. Tsukano: Development of a new in-flight simulator MuPAL-α, *AIAA Modeling and Simulation Technologies Conference*, Denver, USA, AIAA 2000-4574, 2000.

[17] S.S. Niu, G. Fisher: Recursive Information Forgetting with Augmented UD Identification, *International Journal of Control*, 63(3), Feb. 1996.

[18] G. Puyou, P. Ezerzere: Tolerance of aircraft longitudinal control to the loss of scheduling information: toward a performance oriented approach, 7th Symposium on Robust Control Design, Aalborg, Denmark, 2012.

[19] M.E. Salgado, G.C. Goodwin, R.H. Middleton: Modified Least Squares Algorithm Incorporating Exponential Resetting and Forgetting, *International Journal of Control*, 47(2), 1988.

[20] M. Sato, A. Satoh: Flight control experiment of Multipurpose-Aviation-Laboratory-α in-flight simulator, *Journal of Guidance, Control, and Dynamics*, 34(4):1081-1096, 2011.

[21] D. Shore, M. Bodson: Flight testing of a reconfigurable control system on an unmanned aircraft, *Journal of Guidance, Control, and Dynamics*, 28(4):698–707, 2005.

[22] D.G. Ward, J.F. Monaco, M. Bodson: Development and flight testing of a parameter identification algorithm for reconfigurable control, *Journal of Guidance, Control, and Dynamics*, 21(6):698–707, 1998.

[23] Y. Watanabe *et al.*: Vision-integrated navigation system for aircraft final approach in case of GNSS/SBAS or ILS failures, *AIAA GNC*, SciTech Forum, AIAA 2019-0113, San Diego, USA, January 2019.

[24] A. Zolghadri: Advanced model-based FDIR techniques for aerospace systems: Today challenges and opportunities, *Progress in Aerospace Sciences*, 53:18-29, 2012.