JOURNAL OF AIRCRAFT
Vol. 35, No. 3, May–June 1998

Multi-Input/Multi-Output Adaptive Active Flutter Suppression for a Wing Model

M. Andrighettoni* and P. Mantegazza†
Politecnico di Milano, Milano 20133, Italy

The work develops and verifies experimentally a multi-input/multi-output adaptive active flutter suppression system for a built-up wing model fitted with a leading- and a trailing-edge control surface and two accelerometers. An indirect adaptive scheme, combining an identification of a discrete dynamic model followed by the design of a stabilizing control law, has been used. The identification is based on a recursive multivariable-extended least-squares approach applied to the input/output relations connecting the accelerometer signals to the control surfaces deflections, while the stabilizing controller is designed by a full-state eigenstructure assignment method. The order and structure of the control system and the tuning of its design parameters have been carried out through extensive numerical simulations aimed at the determination of an optimal compromise between system performances and computational requirements. The adaptive controller thus obtained was implemented on a wing model and tested in a wind tunnel by using two loosely coupled cooperating personal computers, one for the position servos of the control surfaces, the other for the indirect adaptive flutter suppression. The tests have demonstrated that the system is capable of achieving a significant increase of the subcritical aeroelastic damping and of the flutter speed of the wing model.

Introduction

ACTIVE flutter suppression is a relatively mature but still rewarding research area in aeroservoelasticity. In fact, the need to operate the related active control systems in the presence of a large number of varying configurations and operational parameters, such as velocity, air density, Mach number, air turbulence, and to design them on the basis of relatively uncertain linearized mathematical models governed by complex aeroelastic interactions, makes it very difficult to synthesize an active flutter suppression system having a unique control law, which is effective throughout the entire flight envelope. It is thus often necessary to resort to a scheduling policy that is generally difficult and risky to verify and tune in flight, particularly in the presence of complex control laws.[1] The use of truly adaptive controllers can be a viable alternative solution that is expected to be more flexible, require less in-flight tuning, and also possess a certain degree of adaptiveness in maintaining the aircraft functionality in the presence of operational malfunctions, e.g., control or sensor failures.[2]

However, there are still relatively few applications of adaptive flutter suppression systems, and there are far less consistent design techniques for scheduled flutter suppression than their nonscheduled counterparts. Most of the latest works dealing with a truly adaptive flutter control are related to single-input/single-output aeroservoelastic systems and relatively few have been verified experimentally.[3–6]

The preceding works are mainly based on an indirect adaption scheme in which the flutter controller is designed after explicitly identifying an appropriate representation of the system dynamics. Thus, Ref. 3 characterizes the system through the identification, either by a least-squares or by a maximum likelihood method, of a simplified transfer function that is used to design a stabilizing control law for a single possible flutter mode; whereas Ref. 4 uses an autoregressive moving average with an exogenous input (ARMAX) least-squares identifier to determine the system step-response matrix upon which a proportional feedback is designed.

Along the same lines, Ref. 5 designs an optimal controller based on the explicit identification of the response equation by using various identification techniques. A somewhat different approach is preferred in Ref. 6, which adopts an LMS feedforward scheme, similar to those used in active noise suppression, to generate a counterinstability signal to control the flutter. The present paper pursues the objective of developing an implicit adaptive multi-input/multi-output (MIMO) flutter suppression system that is robust against unmeasured disturbances, e.g., gusts and nonlinearities, and effective over a wide range of instabilities related to arbitrarily time-varying system parameters. It is the culmination of an extensive simulation work[2] that has shown that the use of an indirect adaptive scheme combining a model identification and a stabilizing controller designed by using a full-state eigenstructure assignment is feasible and robust enough to make it worthwhile for implementation on an existing wing model.

Physical and Numerical Wing Model

The model used is a built-in wing (Fig. 1) of aspect ratio 8.25, taper ratio 2, wing surface 0.31 m^2, root chord 0.365 m, span 1.22 m, with Wortmann FX L V-152 K 25 airfoil sections and two control surfaces ranging from 64 to 90% of the wing span and covering 20 and 25% of the chord, respectively, for the leading- and trailing-edge control surface. The most significant vibration modes and frequencies of the wing, with blocked control surfaces, are listed in Table 1.

Two brushless dc motors, placed at the wing root, actuate the control surfaces through connecting shafts. The motors are controlled by closed-loop current power drives that, independent from the actual motor state, allow a direct control of the actuating torque with a bandwidth greatly exceeding the one envisioned for the position servos. The control of the motor torque T_c is based on a proportional-integral-derivative (PID)

Received Aug. 5, 1996; revision received Oct. 15, 1997; accepted for publication Nov. 26, 1997. Copyright © 1998 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Graduate Engineer, Dipartimento di Ingegneria Aerospaziale, Via C. Golgi 40.

†Professor of Aeroelasticity, Dipartimento di Ingegneria Aerospaziale, Via C. Golgi 40. E-mail: mantegazza@aero.polimi.it.

Table 1 Wing vibration modes and frequencies

Mode	NASTRAN frequency, Hz	Experimental frequency, Hz
First bending	3.07	3.1
In-plane bending	5.95	6.0
First torsion	8.2	8.5
Second bending	18.45	18.6

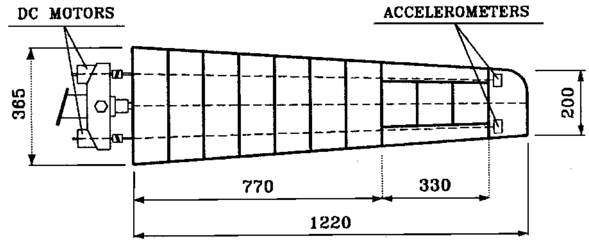

Fig. 1 Sketch of the wing model (measures in millimeters).

command combined with a low-pass second-order shaping filter with passband ω_s and is given by

$$T_c = \frac{as^2 + bs + c}{s(s + A)}\left(\frac{\omega_s^2}{s^2 + 2\xi\omega_s s + \omega_s^2}\delta_c - \delta_a\right) \tag{1}$$

δ_a being the controlled surface deflection, as sensed by potentiometers placed at the wing tip side of each aileron, and δ_c is the related commanded position. The PID compensator is designed by approximating the response of an isolated free control surface, as given by $s^2 J_a \delta_a = T_c$, J_a being the moment of inertia around the hinge axis so that its closed-loop response is governed by

$$\delta_a = \frac{as^2 + bs + c}{(J_a s^4 + J_a A s^3 + as^2 + bs + c)} \times \frac{\omega_s^2}{(s^2 + 2\xi\omega_s s + \omega_s^2)}\delta_c \tag{2}$$

The parameters a, b, c, and A are determined by prototyping the denominator, i.e., the closed-loop poles, of the first term of Eq. (2) according to a fourth-order, low-pass Thomson filter,[7] whereas the command preshaping passband is chosen in such a way to avoid the substantial bandwidth increase and significant step response overshoot caused by the zeros of the closed-loop transfer function. The wing model is completed by two piezoresistive accelerometers mounted at the wing tip, close to the control surfaces hinges, whose signal are antialiased by using two second-order Butterworth filters.

The wing response equations, to be used in the design simulations, have been determined by exploiting the aeroservoelastic modeling capabilities of NASTRAN, that allow the inclusion of arbitrary transfer functions to connect the aeroelastic response to servos, sensors, and related filters. This leads to an equation of the type

$$(s^2[M_g] + s[C_g] + [K_g] - q[H_{\mathrm{am}}(k, M)])\{\boldsymbol{q}_g\} = [B_c]\{\boldsymbol{\delta}_c\} \tag{3}$$

where $s = \sigma + j\omega$ is the complex frequency; and $[M_g]$, $[C_g]$, and $[K_g]$ are the generalized mass, damping, and stiffness matrices related to the generalized degrees-of-freedom vector $\{\boldsymbol{q}_g\}$, of order 12, including the four base structural modes previously described, the two control surfaces rotations, the PIDs dynamics, the command shaping, and the accelerometers antialiasing filters. The aerodynamic transfer matrix $[H_{\mathrm{am}}(k)]$, with $k = (\omega \cdot c)/2V$, c is the mean aerodynamic chord, and V, the asymptotic wind speed, is scaled by the dynamic pressure $q = \frac{1}{2}\rho V^2$ and is available in tabulated form at different reduced frequencies and, because of the low operational wind speeds, is assumed independent from the Mach number.

Equation (3) is not amenable for the simulations emulating the real-time response to be carried out during the design phase, and so a formulation of the unsteady aerodynamic forces in the time domain must be found.

To this end a finite state's approximation[8] could be used, but we prefer the simpler approach of setting

$$[H_{\mathrm{am}}(k)] \cong k^2[M_a] + jk[C_a] + [K_a] \tag{4}$$

the matrices $[M_a]$, $[C_a]$, and $[K_a]$ being obtained through a least-squares fit of the available $[H_{\mathrm{am}}(k)]$ which, to cover the range of frequencies related to the flutter onset, extend up to $k = 2$. Thus, despite its appearance, Eq. (4) is not to be taken as a quasisteady formulation of the aerodynamics, as the latter implies an approximation of $[H_{\mathrm{am}}]$ at low reduced frequencies through a second-order power-series expansion of $[H_{\mathrm{am}}(k)]$ around $k = 0$ for which

$$[M_a] = \frac{d^2[H_{\mathrm{am}}(0)]}{d(jk)^2} = \frac{d^2[\mathrm{Re}(H_{\mathrm{am}}(0))]}{d(k)^2}$$

$$[C_a] = \frac{d[H_{\mathrm{am}}(0)]}{d(jk)} = -\frac{d[\mathrm{Im}(H_{\mathrm{am}}(0))]}{dk}$$

$$[K_a] = H_{\mathrm{am}}(0) = \mathrm{Re}(H_{\mathrm{am}}(0))$$

In practice, such an approximation can be carried out by determining $[M_a]$, $[C_a]$, and $[K_a]$ with a precise fit of Eq. (4) over a few, i.e., 2–3, very small values of k. Thus, Eq. (4) has no particular physical meaning for the application at hand, but is only a simple model that will be shown to be adequate for predicting a relatively high reduced frequency flutter with an acceptable precision.

It allows one to directly approximate Eq. (3) in the time domain with

$$([M_g] - q(c/2V)^2[M_a])\{\ddot{\boldsymbol{q}}_g\} + ([C_g] - q(c/2V)[C_a])\{\dot{\boldsymbol{q}}_g\} + ([K_g] - q[K_a])\{\boldsymbol{q}_g\} = [B_c]\{\boldsymbol{\delta}_c\} \tag{5}$$

so that, after defining $[M_{\mathrm{AE}}] = [M_g] - q(c/2V)^2[M_a]$, $[C_{\mathrm{AE}}] = [C_g] - q(c/2V)[C_a]$, $[K_{\mathrm{AE}}] = [K_g] - q[K_a]$, it can be readily put into the following state form to be used for the simulations:

$$\begin{Bmatrix} \{\dot{\boldsymbol{q}}_g\} \\ \{\ddot{\boldsymbol{q}}_g\} \end{Bmatrix} = \begin{bmatrix} [0] & [I] \\ -[M_{\mathrm{AE}}]^{-1}[K_{\mathrm{AE}}] & -[M_{\mathrm{AE}}]^{-1}[C_{\mathrm{AE}}] \end{bmatrix} \begin{Bmatrix} \{\boldsymbol{q}_g\} \\ \{\dot{\boldsymbol{q}}_g\} \end{Bmatrix} + \begin{bmatrix} [0] \\ [M_{\mathrm{AE}}]^{-1}[B_c] \end{bmatrix} \{\delta_c\} + \begin{bmatrix} [0] \\ [B_n] \end{bmatrix} \{\boldsymbol{n}\} \tag{6}$$

The noise term $\{\boldsymbol{n}\}$ is added to verify the adaptive flutter suppression system behavior in relation to realistic external disturbances, e.g., turbulence and control noise.

A V–g analysis did not show any substantial discrepancy between the eigenvalues of Eq. (6) and those of a p–k solution of Eq. (3), thus demonstrating that Eq. (6) is an adequate model for the prediction of the flutter condition. This has been the only verification carried out to validate the approximation introduced by Eq. (4) as the actual adaptive control should be quite insensitive to the model used to numerically simulate its behavior in the preliminary design phase; otherwise it will be deemed to failure during its implementation.

Identification Methods

Reference 2 has shown that either an extended Kalman filter (EKF) or a recursive least-squares (RLS) approach could be used in the identification phase of the indirect adaptive system. While the EKF proved itself very effective and far superior to the RLS approach, the related computational load was excessive for an implementation on a low-cost personal computer. In this work it is verified that an extended least-squares (ELS) formulation can be a reasonable compromise in terms of control effectiveness and computational load.

Thus, the methods adopted for the identification of the model to be used to design the adaptive active flutter suppression system (AAFSS) are all based on a recursive multivariable autoregressive moving average with exogenus variable (ARMAX) matrix formulation of the type[9]

$$\{\boldsymbol{y}_k\} = \sum_{i=1}^{n_a} [Y_i]\{\boldsymbol{y}_{k-i}\} + \sum_{i=1}^{n_b} [U_i]\{u_{k-i}\} + \sum_{i=1}^{n_c} [\Xi_i]\{\boldsymbol{\xi}_{k-i}\} + \{\boldsymbol{\xi}_k\} \tag{7}$$

where n_a, n_b, and n_c define the order of the model; $[Y_i]$, $[U_i]$, and $[\Xi_i]$ are unknown matrices; $\{\boldsymbol{y}_k\}^T = \{\boldsymbol{y}_k^1 \cdots \boldsymbol{y}_k^{m_a}\}$ is the output vector of order m_a; $\{\boldsymbol{u}_k\}^T = \{\boldsymbol{u}_k^1 \cdots \boldsymbol{u}_k^{m_b}\}$ is the input vector of order m_b; and $\{\xi_k\}^T = \{\boldsymbol{\xi}_k^1 \cdots \boldsymbol{\xi}_k^{m_a}\}$ is the unmodeled/unmeasured white noise vector forcing the system at time k.

It is important to remark that the polynomial matrices $[\Xi_i]$ imply a stable noise-shaping process generating the actual system disturbances from a white noise, and that the stability of such a process must be guaranteed to assure a stable identification.

The preceding relations can be synthesized in the following standard form[9]:

$$\{\boldsymbol{y}_k\} = [\vartheta]\{\boldsymbol{\varphi}_k\} + \{\boldsymbol{\xi}_k\} \tag{8}$$

with

$$\{\boldsymbol{\varphi}_k\} = \{\{\boldsymbol{y}_{k-1}\}^T\{\boldsymbol{y}_{k-2}\}^T \cdots \{\boldsymbol{y}_{k-n_a}\}^T$$
$$\{\boldsymbol{u}_{k-1}\}^T\{\boldsymbol{u}_{k-2}\}^T \cdots \{\boldsymbol{u}_{k-n_b}\}^T \quad \{\boldsymbol{\varepsilon}_{k-1}\}^T\{\boldsymbol{\varepsilon}_{k-2}\}^T \cdots \{\boldsymbol{\varepsilon}_{k-n_c}\}^T\}^T$$
$$[\vartheta] = \begin{bmatrix} \{\boldsymbol{\vartheta}_1\}^T \\ \{\boldsymbol{\vartheta}_2\}^T \\ \cdots \\ \{\boldsymbol{\vartheta}_{m_a}\}^T \end{bmatrix}$$
$$\{\boldsymbol{\vartheta}_i\}^T = \{\boldsymbol{a}_{i,1}^{(1)} \cdots a_{i,1}^{(m_a)} \cdots a_{i,n_a}^{(1)} \cdots a_{i,n_a}^{(m_a)}$$
$$b_{i,1}^{(1)} \cdots b_{i,1}^{(m_b)} \cdots b_{i,n_b}^{(1)} \cdots b_{i,n_b}^{(m_b)} \quad c_{i,1}^{(1)} \cdots c_{i,1}^{(m_c)} \cdots c_{i,n_c}^{(1)} \cdots c_{i,n_c}^{(m_c)}\} \tag{9}$$

and the prediction error $\{\boldsymbol{\varepsilon}_k\} = \{\boldsymbol{y}_k\} - [\vartheta]\{\boldsymbol{\varphi}_k\}$ is assumed as an estimate of $\{\boldsymbol{\xi}_k\}$.

Two methods have been envisaged to identify the system in presence of time-varying parameters. The first is based on a standard recursive extended least-squares (RELS) approach with forgetting factor and covariance bias to account for time-varying parameters.[9]

The second adopts a moving window (MWELS)[2] that applies the ELS approach to the last N time samples so that the system parameters are evaluated by solving the canonical normal least-squares equations by recursively updating the Cholesky factor of the coefficient matrix and right-hand side, through the addition of the contribution at time k while subtracting the factors related to time k-N.[10]

The preceding approaches can be reduced to an RLS formulation by simply dropping the $[\Xi]$ matrices. As already noted, the RLS approach can substantially reduce the computational cost, but, in the presence of significant unmeasurable external disturbances, leads to a biased estimate of the system parameters that can, in turn, produce unstable control laws while the ELS results approach those obtainable with an EKF without requiring excessive computational resources. To be used in the full-state eigenstructure assignment design the input output matrix relation, i.e., (Eq. 7), is finally set into the following state form:

$$\begin{aligned} \{\boldsymbol{x}_k\} &= [F]\{\boldsymbol{x}_{k-1}\} + [G]\{\boldsymbol{u}_{k-1}\} + [E]\{\boldsymbol{\xi}_{k-1}\} \\ \{\boldsymbol{y}_k\} &= [H]\{\boldsymbol{x}_k\} + \{\boldsymbol{\xi}_k\} \end{aligned} \tag{10}$$

with

$$\{\boldsymbol{x}_k\} = \begin{Bmatrix} \{\boldsymbol{y}_k\} \\ \{\boldsymbol{y}_{k-1}\} \\ \cdots \\ \{\boldsymbol{y}_{k-n+1}\} \end{Bmatrix}, \quad [F] = \begin{bmatrix} [Y_1] & [I] & 0 & \cdots & 0 \\ [Y_2] & 0 & [I] & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ [Y_n] & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$[G] = \begin{bmatrix} [U_1] \\ [U_2] \\ \cdots \\ [U_n] \end{bmatrix}, \quad [E] = \begin{bmatrix} [\Xi_1] \\ [\Xi_2] \\ \cdots \\ [\Xi_n] \end{bmatrix}, \quad [H] = [[I]\ [0] \cdots [0]]$$

where, for sake of simplicity, it is assumed that $n = n_a = n_b = n_c$, as it will be in our application.

Controller Design

The design of the stabilizing flutter controller will be based on an eigenstructure assignment method that tries to optimize system performances with an acceptable computational load. It is noted that, once the system controllability can be ensured from any of the available inputs, the availability of the full state, as inferred from Eq. (10), and of two controls allows some redundancy in the eigenvalue assignment. We will exploit that by determining a proportional control law

$$\{\boldsymbol{u}_k\} = [K]\{\boldsymbol{x}_k\} \tag{11}$$

in such a way that the eigenstructure of the closed-loop system matrix ($[F] + [G][K]$) achieves a set of imposed eigenvalues while matching a desired set of corresponding eigenvectors as closely as possible.

To this end, for each assigned eigenstructure, we optimize the following performance index:

$$\begin{aligned} P_i = {} & (\{\boldsymbol{X}_{d_i}\} - \{\boldsymbol{X}_i\})^T[W](\{\boldsymbol{X}_{d_i}\} - \{\boldsymbol{X}_i\}) \\ & + \{\boldsymbol{\lambda}_l\}^T[(s_{d_i}[I] - [F])\{\boldsymbol{X}_i\} - [G]\{\boldsymbol{V}_i\}] \end{aligned} \tag{12}$$

where $\{\boldsymbol{X}_{d_i}\}$ is the ith desired eigenvector, $\{\boldsymbol{X}_i\}$ is the ith actual eigenvector, $[W]$ is a positive definite diagonal matrix that weighs the relative importance of not matching different components of the desired eigenvectors, $\{\boldsymbol{\lambda}_i\}$ is the ith Lagrangian multiplier associated to the ith imposed eigenvalue s_{d_i}, and to the related desired eigenvector, $\{\boldsymbol{V}_i\}$ is an intermediate unknown vector defined by $\{\boldsymbol{V}_i\} = [K]\{\boldsymbol{X}_i\}$.

After imposing the stationary condition to Eq. (12) we obtain the following sets of linear symmetric equations:

$$\begin{bmatrix} [W] & [0] & (s_{d_i}[I] - [F])^T \\ [0] & [0] & -[G]^T \\ (\bar{s}_{d_i}[I] - [F]) & -[G] & [0] \end{bmatrix} \begin{Bmatrix} \{\boldsymbol{X}_i\} \\ \{\boldsymbol{V}_i\} \\ \{\boldsymbol{\lambda}_l\} \end{Bmatrix} = \begin{Bmatrix} [W]\{\boldsymbol{X}_{d_i}\} \\ \{\boldsymbol{0}\} \\ \{\boldsymbol{0}\} \end{Bmatrix} \tag{13}$$

from which, after solving for all the unknown $\{\boldsymbol{X}_i\}$ and $\{\boldsymbol{V}_i\}$, the gain matrix $[K]$ can be determined from the following relation:

$$[K] = [V][X]^{-1} \tag{14}$$

with $[V]=[[V_1]\ [V_2]\ \cdots\ [V_n]]$ and $[X]=[[X_1]\ [X_2]\ \cdots\ [X_n]]$ given by

$$[V_i]=\{V_i\}$$

$$[X_i]=\{\boldsymbol{X}_i\}$$

for each real eigensolution and

$$[V_i]=[\{\mathrm{Re}(V_i)\}\ \{\mathrm{Im}(V_i)\}]$$

$$[X_i]=[\{\mathrm{Re}(X_i)\}\ \{\mathrm{Im}(X_i)\}]$$

for each couple of complex conjugate eigensolution.

The solution of each of the Eqs. (13) is carried out by direct substitution, i.e., by first solving the following symmetric linear system:

$$\begin{bmatrix} (s_i[I]-[F])[W]^{-1}(\bar{s}_i[I]-[F])^T & [G] \\ [G]^T & [0] \end{bmatrix} \begin{Bmatrix} \{\boldsymbol{\lambda}_i\} \\ \{\boldsymbol{V}_i\} \end{Bmatrix} = \begin{Bmatrix} (s_i[I]-[F])\{X_{d_i}\} \\ \{\boldsymbol{0}\} \end{Bmatrix} \tag{15}$$

after which the solution is completed for the actual ith eigenvector, i.e.,

$$\{\boldsymbol{X}_i\}=\{\boldsymbol{X}_{d_i}\}-[W]^{-1}(\bar{s}_i[I]-[F])^T\{\boldsymbol{\lambda}_i\} \tag{16}$$

The recovery of the actual eigenvectors is required because the desired eigenvectors will be assigned in such a way that at each control time step they should remain close to those obtained at the previous step. It must be noted that for any unmodified eigenvalues the corresponding $\{\boldsymbol{X}_i\}$ is the same as that of the uncontrolled system and the corresponding $\{\boldsymbol{V}_i\}=0$. Thus, Eqs. (13), (15), and (16) must be solved only for the unsatisfactory eigensolutions.To reduce the control activity we will then attempt a change of the closed-loop response that is as small as possible. Therefore, the eigenvalue assignment criteria is that the identified eigenvalues are kept unchanged when their continuous time equivalents show an acceptable damping factor; otherwise the assigned eigenvalue is chosen in such a way to maintain its frequency while the ratio of the real to the imaginary part has to attain an assigned acceptable value, say 0.15.

Preliminary Verification Through Numerical Simulations

Even if an adaptive control avoids much of the burdensome work needed for implementing a nonadaptive controller, there remains numerous important design decisions that require an accurate preliminary verification before performing tests on the real hardware. Thus, on the base of the numerical model previously presented, extensive numerical simulations were carried out to determine the most significant control parameters, while satisfying the constraints related to the available computational power. The simulations also mimicked the actual control software to ensure its correctness. They were implemented on the personal computer that had to be used in the actual control system and were organized to simulate the real working conditions, including all of the scaling/conversion factors and control saturation, except for the analog input-output (IO) that were simulated only in relation to the acquisition and output conversion delays. Clearly, while the flutter control computer must be powerful enough to satisfy the scheduling requirements of the control system, it is not required to run the simulations in true real time, but only the time synchronization implied by the control system must be satisfied. Because the control surfaces servos were to be implemented on a different control computer, without excessive constraints on their control frequency, a continuous time formulation of their response was used during the simulations. In this way it was possible to debug the digital flutter control program in such a way that, after including the simple analog-to-digital-to-analog conversion software, it was ready for use. Moreover, an accurate timing allowed the determination of a unique safely achievable control and identification frequency and to verify that the control computer could satisfy the difficult real time scheduling requirements. The extensive simulations carried out to define the best design compromise resulted in the following facts:

1) A servobandwidth of at least 20 Hz—this can be obtained by designing a PID with a 20-Hz bandwidth for both the Thomson and command shaping filter. Even if they produced better results, higher bandwidths were discarded to maintain a realistic constraint on the servo in relation to the structural frequencies to be controlled.

2) A short learning phase resulted, with imposed binary random perturbations of about one-degree amplitude on both control surfaces, to ensure a stable identification in closed loop below the critical speed, the imposed perturbation was not required beyond the critical speed.

3) A recursive-extended least-squares formulation—in fact, as expected from Ref. 2, the use of the simpler least-square method allows both higher control frequencies and model orders but does not appear capable of reliably increasing the flutter speed. In this view it should also be noted that the moving window approach showed itself equivalent to the use of a corresponding forgetting factor but was discarded because it was computationally more expansive.

4) At least an eighth-order aeroservoelastic model is needed to guarantee a substantial increase of the flutter speed.

5) A unique control and identification frequency ranging from 50 to 70 Hz was the best compromise, for an eighth-order model, in relation to the computational power required and to the robustness of the controller against unmodeled disturbances.

6) A one-step-ahead predictive control, because, with the preceding frequencies, a complete adaption and control cycle was needed for the calculations, so that the imposition of an excessively delayed command was worse than a step-ahead prediction.

7) The level of the unmodeled disturbances turned out to be a significant factor in determining the achievable performances, as excessive disturbances required a small forgetting factor; otherwise an unstable controller resulted because of the identification noise. Only smooth time variations of the system parameters are likely to be tolerated in an actual implementation. Clearly, the preceding conclusions were dependent from each other, e.g., a higher-order model could improve system performances, but the limited computer power impeded its implementation at an acceptable frequency.

Finally, the simulations pointed out that the disturbance-shaping process can become unstable, particularly during its initialization phase in a cold start of the identification procedure. Thus, the stabilization of the noise-shaping matrices of Eq. (7) is essential in maintaining a stable identification.

To impose and maintain such a stability, the following equivalent state model of the white noise shaping filter is set up

$$\{\boldsymbol{n}_k\}=[P]\{\boldsymbol{n}_{k-1}\}+[Q]\{\boldsymbol{d}_{k-1}\} \tag{17}$$

with

$$[P]=\begin{bmatrix} -[\Xi_1] & -[\Xi_2] & \cdots & \cdots & -[\Xi_n] \\ [I] & [0] & \cdots & \cdots & [0] \\ [0] & [I] & \cdots & \cdots & [0] \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ [0] & [0] & \cdots & [I] & [0] \end{bmatrix}$$

$$[Q]=\begin{bmatrix} [I] \\ [0] \\ \cdots \\ \cdots \\ [0] \end{bmatrix}$$

so that stability can be assured once the absolute value of any eigenvalue of $[P]$ is less than one.

To this end an eigenstructure assignment approach of the type previously presented has been adopted. It consists of calculating the eigenvalues of the matrix $[P]$, and if any of them has an absolute value greater than 1 then all of the eigenvectors of $[P]$ are also computed. The eigensolution assignment technique previously presented is then used to compute a gain matrix $[K_c]$ in such a way that $([P] + [Q][K_c])$ is stable. To this end all of the computed eigenvectors are imposed along with their stable eigenvalues while the unstable ones are changed by maintaining their real to imaginary part ratio while reducing the corresponding absolute value to an assigned positive number slightly less than one, say 0.95. Because of the structure of Eq. (17) and the exact maintenance of the eigenvectors, the new filter matrix $([P] + [Q][K_c])$ has the same structure as $[P]$ and the new $[\Xi]$ matrices can be easily determined. They are related to a stable filtering process close to its unstable parent, so that the convergence of the filter parameters is not changed significantly. The previous procedure adds a substantial cost to the identification and is responsible for an important part of the limitations imposed on the acceptable adaption and control frequency and model order.

Experimental Setup

The experimental setup is based on two personal computers, a 100 MHz-Pentium PC for the AAFSS and a 33 MHz-486 PC for the servos implementation, data acquisition, and overall test supervision. The need for two computers is related to the computational power required for the AAFSS and to the differing control frequencies of the aileron servos and AAFSS. Moreover, for a prompt and low-cost implementation of the system it was decided to use a single data acquisition board placed on the servo control computer and to connect and synchronize the two systems through their standard parallel ports. Thus, the two controllers run completely asynchronously and the need to share some time for synchronization and communications has determined the actual control frequencies of the two systems. In fact, the flutter suppression computer was run at an indirect adaption frequency of 60 Hz, i.e., model identification plus eigenstructure assignment, instead of the maximum possible of 70 Hz determined during the simulations, while the analog design of the servo was put into the digital form through a simple Tustin transform and run at a sampling frequency of 300 Hz, for a nominal servo bandwidth of 20 Hz. The servos were implemented and tested alone against a wind-tunnel speed of up to 37 m/s, during which it was verified that a minimum sampling frequency of 250 Hz led to good performances and the maximum sustained control frequency, i.e., with the computer dedicated just to their control, could reach 900 Hz. The actual chosen frequency was set at 300 Hz as the computer had to guarantee most of the interfacing and monitoring functions for the whole system. At 300 Hz the actual delay between the control interrupt and the corresponding settled analog control output is about one-fifth of the total control time, i.e., 1/4500 s, and includes the potentiometer signal acquisition, the computation of the incremental control action, and its saturation. The remaining time is devoted to the controller update required to prepare for the immediate addition of the increments related to the next control step.

The coordination of the two controls is not tight and is based on the request of the AAFSS at each of its 60 Hz interrupts as follows:

1) The predicted control positions available from the previous control step are sent to the control surfaces computer as the actual positions to be servoed.

2) The servos computer accepts the control commands, but will use them at the next sampling, immediately reads the accelerometers, and sends the corresponding values to the AAFSS computer.

Thus, a control surface command can be delayed a maximum of either (1/300 + 1/4500) s, when it is received just after the ending of the servo control cycle, or at 1/15,000 s, i.e., the time required to receive the command, acquire the accelerometers signals, and send them back to the AAFSS computer, when the latter has interrupted the servocontrol in the idle phase prior of serving its interrupt. The worst implications of this implementation is a delay of up to 20% of the flutter control timing in closing the adaptive flutter loop and up to a 2% delay in the timing of the servoloop. A tighter control action would have required an independent acquisition for each controller, a synchronization based on a unique clock, and a matching of the control frequencies of the two systems. While this was not possible with the available hardware, it is questionable that such an implementation could be acceptable in any operating flying flutter suppression system, as it imposes severe constraints when compared to a decentralized loosely coupled implementation of the type used in this work. In fact, any really flying AAFSS is likely to use an independent acquisition and digital control surface servo based on processors, which, even in the lowest performance range, would be capable of a control rate at least an order of magnitude higher than the one used in this work so that the previous delays will be completely negligible. It is recalled that a substantially higher rate could be achieved for the servos by using a computer of the same class as that used for the flutter suppression. In fact, it has already been pointed out that tripled servo control frequency was possible with the computer at hand, but could not be adopted because of the need to use the related computer as a supervisor of the test activity.

To complete the description of the test setup we note that the acquisition was based on a multiplexed analog input board with a single 100 kHz analog to digital converter and on two digital to analog converters with correspondingly fast overall settling times. The motors, accelerometer, antialiasing filters, and the other design parameters were just as those described in the modeling and simulations paragraphs.

Experimental Verification

An indirect adaptive control system, by explicitly separating the identification from the control phase, allows an easy verification of the stability of the aeroelastic response, at least below the uncontrolled flutter speed. A reliable identification technique to be used in an AAFSS could be a valuable tool also during any experimental flutter program. The first part of the tests was dedicated to the identification of the V–g plot to verify how it correlates to that of the numerical analyses. To this end at each wind-tunnel speed the system is excited by two independent small binary random commands applied to the control surfaces, the eighth-order model to be used for the flutter suppression is identified, and the eigenvalues of the matrix $[F]$ of Eq. (10) are calculated and converted to their equivalent principal values in the continuous time domain.

In Fig. 2 the experimentally identified V–g plot is compared to those available from a p–k analysis of the numerical flutter model and from the simulated tests carried out during the preliminary design phase. It can be seen that a relatively violent flutter at a speed between 25 and 26 m/s can be inferred, which well matches the one computed from the simulations. The cloud of points at each test speed is related to the variation of the identified eigenvalues at constant wind speed and is a clear indication of the need of a robust control design capable of coping with the evidenced uncertainties. On the other hand, because any interextrapolation of their values at changing speed is difficult, it gives a clear indication of the uncertainties that make flutter tests a risky exploration of the unknown. In this view it is important to remark that the identification leads to an overall good matching of the important system frequencies, while the damping trend is very good for the most important mode, i.e., first torsion, that leads to the flutter, while only qualitatively matching that of the first bending mode that

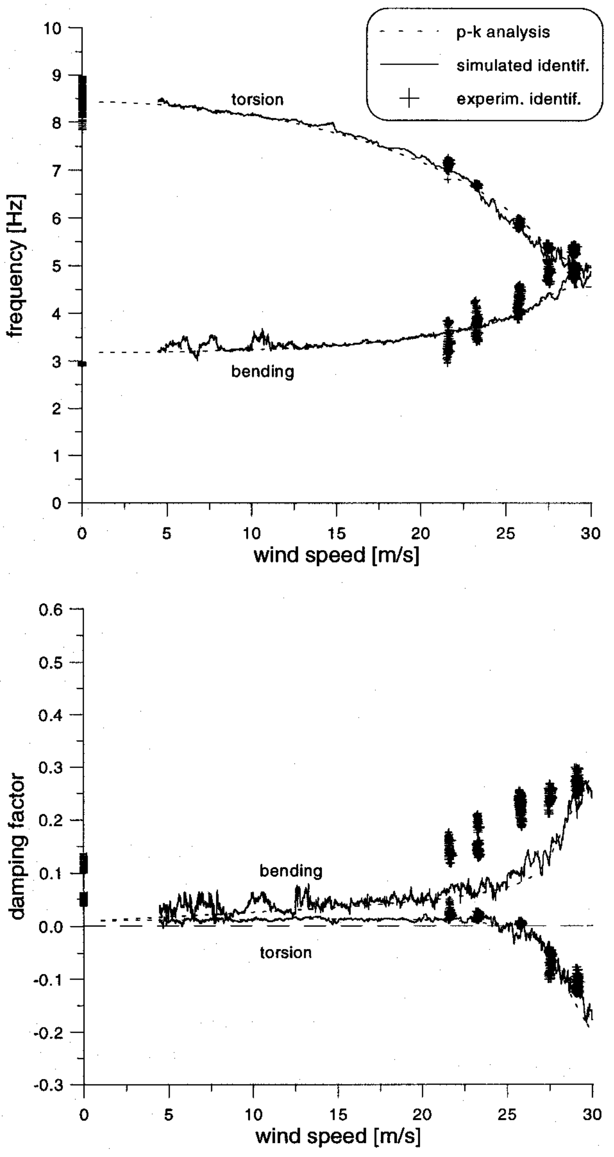

Fig. 2 Comparison of different computed and identified V–g plots.

is increasing as the speed gets higher, so that its participation to the overall motion is increasingly difficult to identify.

In relation to the critical points evidenced by the simulations the main objectives of the experimental activity were set as follows:

1) Verify the effectiveness of the adaptive system in increasing the damping of the aeroservoelastic system at subcritical speeds.

2) Demonstrate the capability of the adaptive system to suppress flutter at speeds beyond the uncontrolled flutter speed in the presence of significant external perturbations related to wind-tunnel turbulence, substantial controls actuation (±15 deg), and other unmodeled forcing terms, including abrupt torsional and bending deformations, shutting off the control system to activate a well-developed visible unstable response, and then recovering stability by activating the controller again.

3) Determine the maximum achievable flutter speed for an experimentally linearized test condition, i.e., in the presence of relatively small external disturbances such as those related only to the wind-tunnel turbulence.

4) Demonstrate a capability to adapt, up to the controlled flutter speed, to relatively soft-time varying conditions, e.g., the wind-tunnel startup.

The following figures show the extent to which these objectives were satisfied. In particular, Fig. 3 demonstrates the capability of adding damping up to the uncontrolled flutter speed, whereas Fig. 4 shows that, well beyond the flutter speed, the active suppression is maintaining the wing stable with a reasonable control activity after successfully recovering

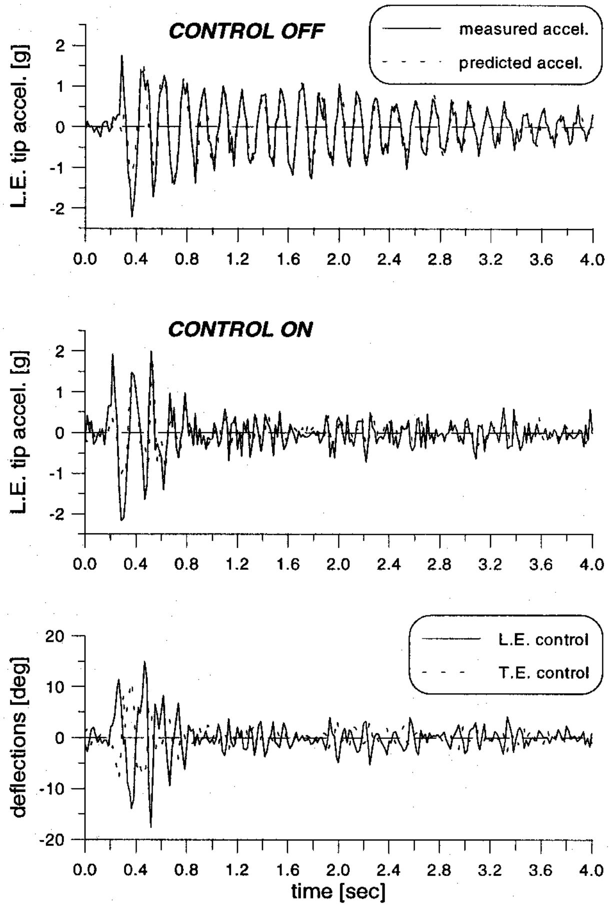

Fig. 3 Effectiveness of the adaptive flutter control in improving the damping while approaching the uncontrolled flutter onset (25 m/s).

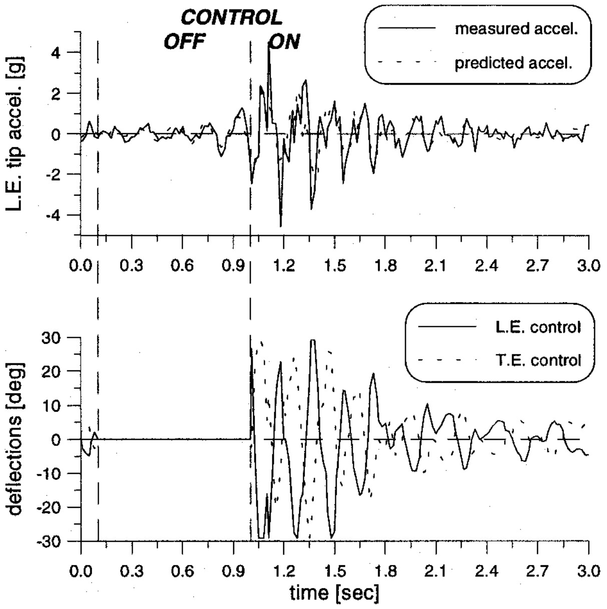

Fig. 4 OFF/ON recovery for a controlled flutter condition (30.2 m/s).

the stability from a fully developed violent flutter caused by shutting off the controller. A justification for the large control motions in the first phase of the recovery will be given later in this paper.

Figure 5 proves that the adaption is still working when the system parameters are varying because of two fast transition steps of the wind-tunnel speed, followed by relatively long steady operational conditions to verify the maintenance of stability, during a wind-tunnel startup to reach a speed 20% beyond that of the uncontrolled flutter.

The results shown are those related to a portion of what appeared as the most critical flutter recovery conditions from substantial external perturbations. Similar results could be obtained after the imposition of strong control surfaces jerks and abrupt deformations, both impulsively and in steps.

From them, and a wealth of other experiments, it can be stated that the controller can increase the flutter speed up to 30.5 m/s. Under relatively smaller perturbations, like those related only to the wind-tunnel turbulence, the model remained stable up to 33 m/s with a marginal capability to maintain stability against milder perturbations, as demonstrated by Fig. 6. However, beyond 30.5 m/s it was not possible to shut the controller off, with a sure recovery from flutter when it was back on, or to significantly perturb the system by other means with a guaranteed recovery of a stable response. The reason for such a loss of effectiveness can be inferred from Fig. 6, where it can be seen that the average control activity is of a few degrees, except when strong turbulence bursts are encountered. In such cases, as well as when very large perturbations are applied (Fig. 4), the ailerons are forced to counteract to saturation, so that if further perturbation is added flutter can develop, particularly if the high-level perturbation frame is a long lasting one.

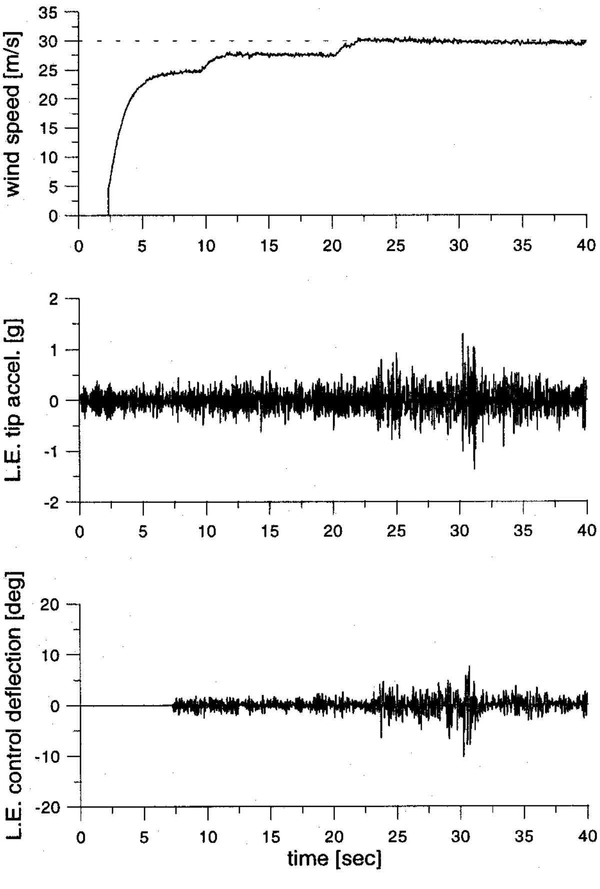

Fig. 5 Adaptive flutter suppression under varying operating conditions related to the wind-tunnel startup (final wind speed of 30.2 m/s).

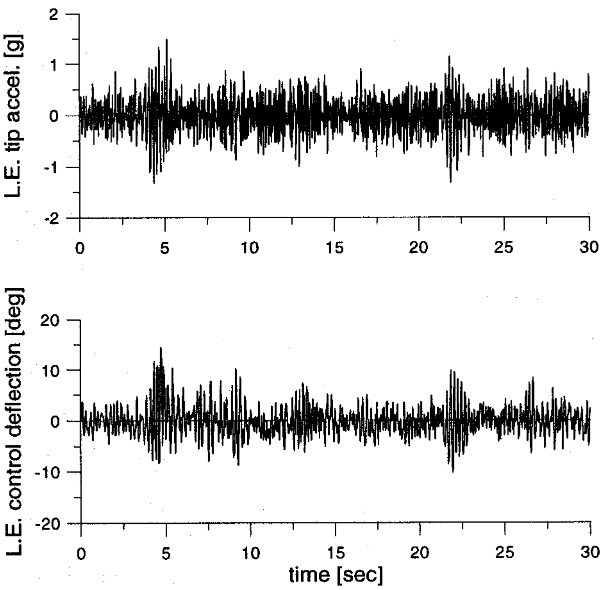

Fig. 6 Adaptive flutter suppression in a steady flow at 33 m/s. Wing excited only by the wind-tunnel turbulence.

The successful results reported can be viewed as partially successful if compared to those of the simulations that predicted a controlled flutter speed of 35 m/s under large disturbances.

A possible plausible justification of this discrepancy can be related to the use of a linear model in the numerical simulations, whereas significant nonlinearities that were mainly related to the aerodynamics appeared in the actual tests. In fact, because of excessively large side gaps and an inaccurate blending, a significant flow detachment at the sides of the control surfaces was visualized during the tests, and about 20% of the control surface length presented a separated flow. Such a condition could be further worsened when, as noted in relation to the results of Figs. 4 and 6, very large control deflections were required to suppress strong disturbances, because in such cases the detached flow at the leading edge caused a sizable portion of the trailing-edge flap to operate largely within the separation wake. The consequent strong nonlinear behavior and loss of control efficiency related to such stalled conditions made the reality more and more differing from the linear simulations, thus justifying a larger than predicted control activity.

Other less important but nonnegligible causes of the partial loss of correlation between the simulations and experiments can be related to the loose synchronization of the control computers, explained in the previous paragraph, that was not accounted in the simulation, and to the larger than expected noise in the accelerometer signals, that deteriorated the identification quality. At the beginning of the experimental activity such a noise seemed completely out of control and related to everything that was connected to an ac power plug. The power line transient related to turning the wind tunnel on made the accelerometers measure nonexistent motions that drove the servos mad. A careful setup of the different ground loops solved most of the problems, but the signal-to-noise ratio of the accelerometers remained worse than expected, so that, although incapable of solving the problem, the tests were still carried out.

Concluding Remarks

This paper has demonstrated that an indirect adaptive active flutter suppression, coupling a multivariable-extended least-squares identification and a robust stabilizing controller based on a full eigenstructure assignment, is viable and capable of significantly extending the flutter envelope of a built-in wing model, both for steady and time varying operational conditions. Even if the adopted approach is basically streamlined along relatively standard techniques, the use of a MIMO adaptive flutter control system and some of its implementation details, e.g., the approximate optimal assignment of a whole eigenstructure, are believed to present some points of originality. The test results are somewhat inferior to those predicted by the linear simulations, particularly when substantial perturbations are applied at the highest speeds permitted by the active flutter suppression, while a behavior closer to that simulated is achieved under strong disturbances for a speed up to 20% beyond the uncontrolled flutter onset. Such a fact has been given a plausible justification in relation to unmodeled aerodynamic nonlinearities, synchronization delays between the independently operating control computers, and measurement noises that could not be improved with the available hardware. In this view it is believed that the level of correlation reached between the simulations and the tests is fairly good and the capability to partially cope with unexpected operational conditions is a demonstration of the robustness of the adopted indirect adaption scheme. Whether the adaptive approach presented in this paper can be a more effective solution with respect to more conventional robust controllers, or other adaptive techniques, is a point that remains to be verified. It is thus believed that, despite the already demonstrated potentialities, there remains many interesting, and hopefully rewarding, research activities before any adaptive active flutter suppression system can be installed on a truly flying vehicle.

References

[1] Ghiringhelli, G. L., Lanz, M., Mantegazza, P., and Ricci, S., "Active Flutter Suppression Techniques in Aircraft Wings," *Integrated Technology Methods and Applications in Aerospace Systems Design*, edited by C. T. Leondes, Vol. 52, Control and Dynamics Series, Academic, San Diego, CA, 1992, pp. 57–115.

[2] Capone, P., and Mantegazza, P., ''Adaptive Control of a Multi-Input Multi-Output Flutter Suppression System,'' *Proceedings of the 13th AIDAA (Italian Association of Aeronautics and Astronautics) Congress*, AIDAA, Rome, 1995, pp. 763–774.

[3] Johnson, E. H., Hwang, C., Joshi, D. S., Harvey, C. A., Huttsell, L. T., and Farmer, M. G., "Adaptive Flutter Suppression—Analysis and Test,'' AGARD Rept. 703, 1982.

[4] Porter, B., Merzougui, T., and Gu, Z. Q., "Design of Adaptive Fast-Sampling Digital Controllers for Active Flutter Suppression," AIAA Paper 92-2105, 1992.

[5] Pak, C.-G., "Adaptive Active Flutter Suppression of Wings in Subsonic and Transonic Flight Regimes," Ph.D. Dissertation, Univ. of California, Los Angeles, CA, 1991.

[6] Danda Roy, I., and Eversman, W., ''Active Flutter Suppression Using MIMO Adaptive LMS Control," AIAA Paper 96-1345, 1996.

[7] Nachtigal, C. L. (ed.), *Instrumentation & Control*, Wiley, New York, 1990.

[8] Morino, L., Mastroddi, F., De Troia, R., Ghiringhelli, G. L., and Mantegazza, P., "Matrix Fraction Approach for Finite-State Aerodynamic Modeling," *AIAA Journal*, Vol. 33, No. 4, 1995, pp. 703–711.

[9] Iserman, R., Lachmann, K.-H., and Matko, D., *Adaptive Control Systems*, Prentice–Hall, New York, 1992.

[10] Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic, London, 1981.