Identification of structural surfaces' positions on an F/A-18 using the subspace identification method from flight flutter tests

M N Beaulieu, S De Jesus Mota, and **R M Botez***
Automated Production Engineering, École de Technologie Supérieure, Montréal, Québec, Canada

The manuscript was received on 13 March 2007 and was accepted after revision for publication on 19 April 2007.

DOI: 10.1243/09544100JAERO219

Abstract: In the current paper, a linear state-space mathematical model, identified from flight flutter tests is presented, to simulate the aeroelastic deflections of specific structural parts of the NASA F/A-18 Active Aeroelastic Wing research aircraft. The flight flutter tests were performed in steady-level flight with Schroeder frequency excitation induced on the aircraft ailerons by an on-board excitation system activated by the pilot. The results of the flight flutter tests were used to generate an aeroelastic model in which the deflections of the specific aircraft surfaces are functions of the control inputs combined with the deflections of other aircraft surfaces. The F/A-18 linear model is conceived as nine third-order multiple input–single output (MISO) models. Each model has nine inputs and one output. The nine inputs are the differential ailerons deflection and the deflections of all the other parts of the aircraft. The output of each model is the structural deflection of a given aircraft structure. The model's parameters are estimated with the subspace system identification algorithm, an efficient non-iterative algorithm that computes the system matrices directly from the input and output data. The model's quality is evaluated by calculating the fit and correlation coefficients between the model's outputs and the outputs from flight flutter test data. Although the fit coefficient results are very good – between 89 and 99 per cent – the correlation coefficient method gave the best results (nearly 100 per cent). Finally, resampled inputs were used to validate the F/A-18 model's robustness. The model's aircraft structure was validated for flutter flight tests at different Mach numbers and altitudes. The estimated linear model fits the flight flutter test data very well. The subspace method is therefore very convenient for model identification from flight flutter tests.

Keywords: flight tests, validation, aeroelasticity, aeroservoelasticity, F/A-18 aircraft

1 INTRODUCTION

This project uses flight flutter test data obtained from the Active Aeroelastic Wing (AAW) Technology Research Program. The aim of the AAW Technology Flight Research Program, initiated in 1996, was to validate an aircraft concept in which a lighter, more flexible wing could be used to improve the overall aircraft performance. A modified F/A-18 full-scale aircraft was the ideal choice for the AAW technology demonstration, with its high-speed flight capacity and its thin and flexible wings equipped with multiple control surfaces. The modifications performed on the F/A-18 aircraft in the AAW Flight Research Program have been given in detail [**1**, **2**].

The current paper presents a model for the structural oscillations of the flexible F/A-18 AAW. The model is built by use of the subspace parameter estimation methods from flight flutter tests. The next section presents a literature review on aeroelasticity, where system identification methods, mainly the subspace method, were used for model identification from flight flutter tests.

The autoregressive moving average method and neural networks theory were used by Sung *et al.* [**3**] to

**Corresponding author: Automated Production Engineering, École de Technologie Supérieure, 1100 Notre-Dame West, Montréal, Québec, Canada H3C 1K3. email: ruxandra@gpa.etsmtl.ca*

identify the flutter behaviour of a transonic wing. The flutter dynamics of a pitch–plunge system subjected to limit cycle oscillations were modelled by Kukreja and Brenner [**4**] using non-linear autoregressive moving average exogenous non-linear models. Silva *et al.* [**5**] identified the dynamics of a flexible wing model using the impulse response method and the Eigensystem realisation algorithm. In addition, an output-error minimization method was performed by LeGarrec *et al.* [**6**] based on a large flexible aircraft.

The subspace method has already been applied in other fields such as fibre optics research [**7**]. In aerospace, the subspace method has been used to identify, from flight flutter tests, the effects of the aircraft's control surfaces' motion on the rigid modes of the F/A-18 AAW [**8**]. In this method, the ailerons were excited by the use of Schroeder frequency sweeps. The accelerations of the aircraft were filtered using a wavelet transform and the aircraft's roll response was identified in both the time and frequency domains [**8**]. In the present study, the subspace identification method is used to identify the structural deflections of the F/A-18 AAW aircraft's surfaces from flight flutter tests.

2 METHODOLOGY

The methodology section of this paper is divided into three sections. The first section presents the flight flutter tests data for the F/A-18 AAW aircraft. The second section presents the structure of the linear model with its inputs and outputs. The last section explains the details of the subspace identification algorithm used to obtain the parameters of the linear model.

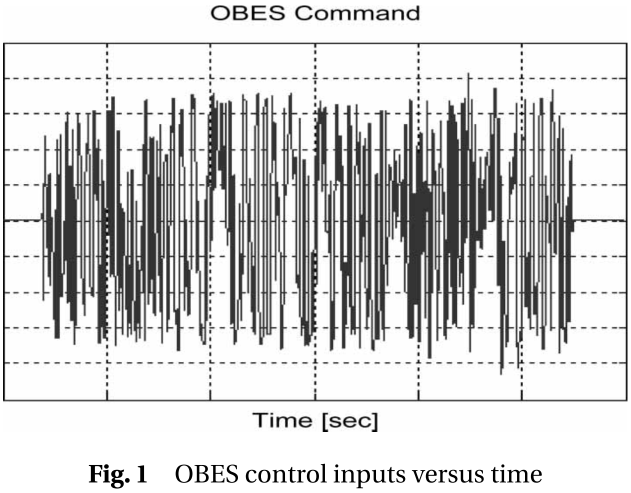

Fig. 1 OBES control inputs versus time

2.1 Flight flutter tests data and their filtering

In order to obtain the recorded flight flutter tests data, the flight control computer (FCC) for the F/A-18 AAW aircraft was modified by adding a research flight control system (RFCS) to generate the Schroeder frequency sweep control inputs. The RFCS processor was activated by the pilot with a cockpit switch, thus adding the actuator commands from the RFCS to the commands from the aircraft's baseline FCC.

The software used by the RFCS to control the actuators was called the on-board excitation system (OBES). The input activated by the OBES was a Schroeder frequency sweep, which is a large number of harmonics, equally spaced in the frequency domain. An example of OBES controls is shown in Fig. 1.

The OBES Schroeder excitation signal is defined in equation (1)

$$\mathrm{OBES}(t) = \sum_{k=1}^{C} A_k \sin(2\pi f_k t + \phi_k) \tag{1}$$

where f_k is the kth measurement frequency, ϕ_k is the kth phase and A_k is the kth amplitude of the OBES Schroeder signal. Details of the theory of Schroeder signals are given by Schroeder [**9**]. The OBES-generated Schroeder signal is sent to the aircraft actuators to generate the F/A-18 control surface oscillations. Records of structural surfaces' accelerations were obtained at 30-s intervals by accelerometers. These tests were performed for a combination of Mach numbers from 0.85 to 1.20 and for altitudes from 5000 to 25 000 ft.

The accelerations from flight flutter test data measured by NASA DRFC laboratories were used. The flight test data accelerations on the structural surfaces are very noisy. The noise was removed in order to identify the F/A-18 AAW model by performing a double integration on the surface accelerations to obtain surface deflections. Since the noise is a random process with a mean zero value, then the integration removes any noise contribution to the data. Therefore, no additional filter is required. Figure 2 shows the schematic of the flight flutter test data preprocessing. The filtering

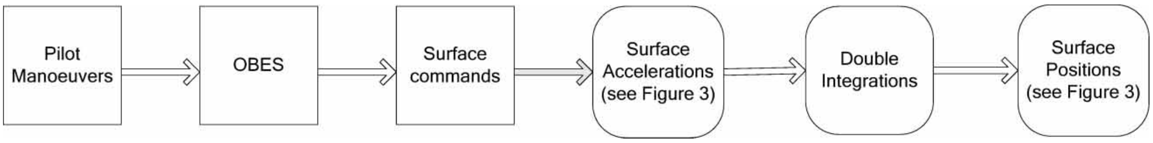
Pilot Manoeuvers
OBES
Surface commands
Surface Accelerations (see Figure 3)
Double Integrations
Surface Positions (see Figure 3)

Fig. 2 Flight flutter tests data preprocessing scheme

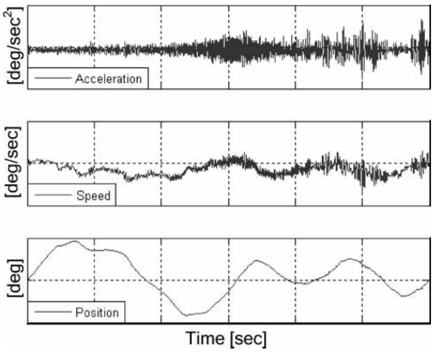

Fig. 3 Left wing accelerations and their integrations with time, which gives the deflection speed and the deflection with time

effect of the acceleration double integration is shown in Fig. 3 where speeds and deflections time histories are presented for the left wing surface.

In Fig. 3, only the structural surface deflection acceleration for the left wing is shown in order to illustrate how the integration operations remove the unwanted noise.

2.2 Linear models

The deflections of the following nine structural surfaces are considered in the current paper:

(a) lwing, rwing – left and right wing;
(b) ltef, rtef – left and right trailing edge flap;
(c) lvert, rvert – left and right rudder;
(d) lstb,rstb – left and right stabilizer;
(e) lstby – left stabilizer lateral.

It was assumed that each structural surface deflection is a function of the left and right ailerons inputs and of the deflection of the other nine aircraft surfaces. The right and left aileron positions are considered the control inputs, and their notations are given below:

(a) lail – left aileron position;
(b) rail – right aileron position.

Nine multiple input–single output (MISO) models are considered in this study, and therefore, there are nine estimated outputs. Figure 4 shows the MISO scheme, where, in this case, the estimated output is the left wing deflection lwing. A similar type of scheme is applied to the other estimated outputs.

In this study, there are always ten inputs associated with the MISO model: nine structural surfaces (as mentioned above), minus one, which is the estimated output (in this case the left wing), plus both right and left aileron positions. Using the method explained above as a model, the other eight MISO models are used to obtain the remaining eight structural surface deflections versus time.

2.3 Subspace system identification algorithm

The state-space matrices parameters (Fig. 4) are obtained by the subspace system identification algorithm, which is described in this section. Generally, a discrete linear model is defined by the following equations

$$\boldsymbol{x}(t+\Delta T)_{n\times 1} = \mathbf{A}_{n\times n}\boldsymbol{x}(t)_{n\times 1} + \mathbf{B}_{n\times m}u(t)_{m\times 1} + \boldsymbol{w}(t)_{n\times 1} \quad (2)$$

$$\boldsymbol{y}(t)_{o\times 1} = \mathbf{C}_{o\times n}\boldsymbol{x}(t)_{n\times 1} + \mathbf{D}_{o\times m}u(t)_{m\times 1} + \boldsymbol{v}(t)_{o\times 1} \quad (3)$$

where t is the time, ΔT is the sample time, $u(t)$ represents the model inputs, $y(t)$ represents the model outputs, m is the number of inputs and o is the number of outputs. In equations (2) and (3), the vector $\boldsymbol{x}(t)$ of length n represents the system states expressed as linear combinations between previous inputs and previous outputs, $\boldsymbol{w}(t)$ represents the state noise vector, and $\boldsymbol{v}(t)$ is the measurement noise vector. The matrices **A**, **B**, **C**, and **D** are well known in modern control theory.

The terms of the **A**, **B**, **C**, and **D** matrices are usually estimated by the use of various parameter estimation methods. Most of these methods start with a set of initial guesses based on physical knowledge of the system. A minimization algorithm is further used to reduce the error between the model output and the given flight test data. Unfortunately, with these methods, if the initial parameter guesses are far from their true values, the minimization algorithm

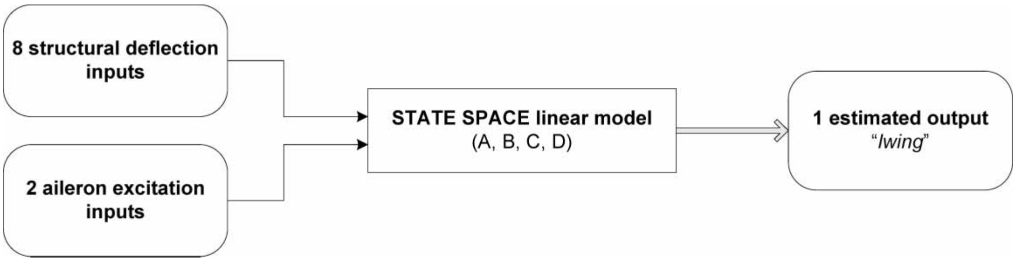

Fig. 4 MISO model with the left wing output

may converge towards a local minimum. The main advantage of the subspace identification algorithm used in this paper is that it is non-iterative and does not require an initial guess of the terms in the matrices [**A**, **B**, **C**, **D**]. The only information required by the subspace method is the input and output data vectors.

The subspace identification algorithm is implemented with the Matlab system identification toolbox. The basic theory of this algorithm is described by Ljung [**10**], and the manner in which the algorithm is implemented in Matlab® is given by Lennart [**11**]. The main concept of the subspace method is the definition of the system observability matrix $\mathbf{\Gamma}_r$ in equation (4) from modern control theory, where r represents a forward prediction horizon [**10**]. This matrix can be obtained from the system inputs $u(t)$ and outputs $y(t)$ and its expression is the following

$$\mathbf{\Gamma}_r \overset{\text{def}}{=} \begin{bmatrix} \mathbf{C}_{o\times n} \\ \mathbf{CA}_{o\times n} \\ \cdots \\ \mathbf{CA}^{r-1}_{o\times n} \end{bmatrix}_{ro\times n} \tag{4}$$

Once this observability matrix $\mathbf{\Gamma}_r$ is known [**10**], then the state space matrices [**A**, **B**, **C**, **D**] are obtained by the use of the input and output vectors.

The detailed procedure to obtain the observability matrix and the discrete state-space matrices will now be explained. The theory behind the subspace algorithm is divided into four sections. Section 2.3.1 describes the basic matrices and equations required for the demonstration, sections 2.3.2 and 2.3.3 explain the two steps necessary to compute the observability matrix of equation (4), and section 2.3.4 explains how the discrete state-space matrices [**A**, **B**, **C**, **D**] are obtained from the observability matrix.

2.3.1 Basic definitions of input and output matrices

In order to understand the subspace algorithm, it is necessary to define a number of important matrices. The input data given later than the reference time t can be arranged into a Hankel matrix as follows

$$\mathbf{U}_{\text{f}} \overset{\text{def}}{=} \begin{bmatrix} u[t]_{m\times 1} & u[t+\Delta t]_{m\times 1} & u[t+2\Delta t]_{m\times 1} & \cdots & u[t+(j-1)\Delta t]_{m\times 1} \\ u[t+\Delta t]_{m\times 1} & u[t+2\Delta t]_{m\times 1} & u[t+3\Delta t]_{m\times 1} & \cdots & u[t+j\Delta t]_{m\times 1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u[t+(r-1)\Delta t]_{m\times 1} & u[t+r\Delta t]_{m\times 1} & u[t+(r+1)\Delta t]_{m\times 1} & \cdots & u[t+(r+j-2)\Delta t]_{m\times 1} \end{bmatrix}_{rm\times N} \tag{5}$$

In this matrix, the subscript f stands for future inputs, because only the inputs given later than time t are included in the matrix. The subscript N represents the length of the output vector from flight flutter tests, and the index j is a dummy variable adjusted such that all data available in the identification are included on each line of the $\mathbf{U}_{\text{f}}$ matrix. Note that if the expression in the brackets of the matrix element u has a value greater than N, the value of $\mathbf{U}_{\text{f}}$ is zero. Similar Hankel matrices can be constructed with the output vectors and defined as $\mathbf{Y}_{\text{f}}$. The future state vector can be defined by the use of the following matrix

$$\mathbf{X}_{\text{f}} = [\mathbf{x}[t]_{n\times 1}\ \ \boldsymbol{x}[t+\Delta t]_{n\times 1}\ \ldots\ \boldsymbol{x}[t+(j-1)\Delta t]_{n\times 1}]_{n\times N} \tag{6}$$

Is also necessary to define the extended controllability matrix

$$\Delta_r = [(\mathbf{A}^{r-1}\mathbf{B})_{n\times 1}\ \ (\mathbf{A}^{r-2}\mathbf{B})_{n\times 1}\ \cdots\ (\mathbf{AB})_{n\times 1}\mathbf{B}_{n\times 1}]_{n\times r} \tag{7}$$

and finally the impulse response matrix is given by

$$\mathbf{H}_r = \begin{bmatrix} \mathbf{D}_{o\times m} & 0 & 0 & 0 & 0 \\ \mathbf{CB}_{o\times m} & \mathbf{D}_{o\times m} & 0 & \cdots & 0 \\ \mathbf{CAB}_{o\times m} & \mathbf{CB}_{o\times m} & \mathbf{D}_{o\times m} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{CA}^{r-2}\mathbf{B}_{o\times m} & \mathbf{CA}^{r-3}\mathbf{B}_{o\times m} & \mathbf{CA}^{r-4}\mathbf{B}_{o\times m} & \cdots & \mathbf{D}_{o\times m} \end{bmatrix}_{ro\times Nm} \tag{8}$$

It is also important to describe the effect of noise on the future output of the system. This noise effect is defined by use of the Hankel matrix **V**

$$\mathbf{V} \overset{\text{def}}{=} \begin{bmatrix} \mathbf{V}_1[t]_{o\times 1} & \mathbf{V}_1[t+\Delta t]_{o\times 1} \\ \mathbf{V}_2[t+\Delta t]_{o\times 1} & \mathbf{V}_2[t+2\Delta t]_{o\times 1} \\ \cdots & \cdots \\ \mathbf{V}_k[t+(r-1)\Delta t]_{o\times 1} & \mathbf{V}_k[t+r\Delta t]_{o\times 1} \end{bmatrix}$$

$$\left.\begin{matrix} \mathbf{V}_1[t+2\Delta t]_{o\times 1} & \cdots & \mathbf{V}_1[t+(j-1)\Delta t]_{o\times 1} \\ \mathbf{V}_2[t+3\Delta t]_{o\times 1} & \cdots & \mathbf{V}_2[t+j\Delta t]_{o\times 1} \\ \cdots & \cdots & \cdots \\ \mathbf{V}_k[t+(r+1)\Delta t]_{o\times 1} & \cdots & \mathbf{V}_k[t+(r+j-2)\Delta t]_{o\times 1} \end{matrix}\right]_{ro\times N} \tag{9}$$

where the value of the term $\mathbf{V}_k$ is defined with equation (10)

$$\begin{aligned} \mathbf{V}_k = {} & \mathbf{CA}^{k-2}\boldsymbol{w}(t) + \mathbf{CA}^{k-3}\boldsymbol{w}(t+1) + \cdots \\ & + \mathbf{C}\boldsymbol{w}(t+k-2) + \boldsymbol{v}(t+k-1) \end{aligned} \tag{10}$$

In the subspace algorithm, the future value of the output is related to the future value of the states by use of the following equations

$$\mathbf{X}_{\mathrm{f}} = \mathbf{A}^r\mathbf{X}_{\mathrm{p}} + \Delta_r\mathbf{U}_{\mathrm{p}} \tag{11.1}$$

$$\mathbf{Y}_{\mathrm{f}} = \boldsymbol{\Gamma}_r\mathbf{X}_{\mathrm{f}} + \mathbf{H}_r\mathbf{U}_{\mathrm{f}} + \mathbf{V} \tag{11.2}$$

Equations (11.1) and (11.2) were derived from state-space equations (2) and (3). The next section will show how the observability term $\boldsymbol{\Gamma}_r\mathbf{X}_{\mathrm{f}}$ can be isolated from equation (11.2).

2.3.2 Removal of input and noise contribution to the outputs

The first step to obtain the observability matrix $\boldsymbol{\Gamma}_r$ is to isolate the term dependant on the states $\boldsymbol{\Gamma}_r\mathbf{X}_{\mathrm{f}}$ in equation (11.2). This procedure can be divided into two steps: (a) an orthogonal projection done to remove the input contribution $\mathbf{H}_r\mathbf{U}_{\mathrm{f}}$ and (b) an instrument variable added to remove the noise contribution $\mathbf{V}$. The algorithm used to perform this procedure is called the instrumental variables approach, thoroughly explained in references [**10**], [**12**], and [**13**], and here briefly described in the following sections 2.3.2.1 and 2.3.2.2.

2.3.2.1 Orthogonal projection to remove the input contribution. To remove the input contribution $\mathbf{H}_r\mathbf{U}_{\mathrm{f}}$, a geometric interpretation of equation (11.2) must be used, as shown in Fig. 5 and explained in reference [**7**].

If equation (11.2) is interpreted as a vector, the output contribution can be removed by projecting the output vector $\boldsymbol{Y}_{\mathrm{f}}$ perpendicular to the input contribution $\mathbf{H}_r\mathbf{U}_{\mathrm{f}}$, which can be achieved with the following projection matrix

$$[\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}}]_{N\times N} = \mathbf{I} - \mathbf{U}_{\mathrm{f}}^{\mathrm{T}}(\mathbf{U}_{\mathrm{f}}\mathbf{U}_{\mathrm{f}}^{\mathrm{T}})^{-1}\mathbf{U}_{\mathrm{f}} \tag{12}$$

where the superscript T means transpose. A detailed proof concerning this orthogonal projection matrix is available in reference [**7**]. If this perpendicular projection operator is applied on $\mathbf{U}_{\mathrm{f}}$, it is equivalent to finding the projection of $\mathbf{H}_r\mathbf{U}_{\mathrm{f}}$ perpendicular to $\mathbf{U}_{\mathrm{f}}$ which, logically, is zero. Mathematically, it can be shown as

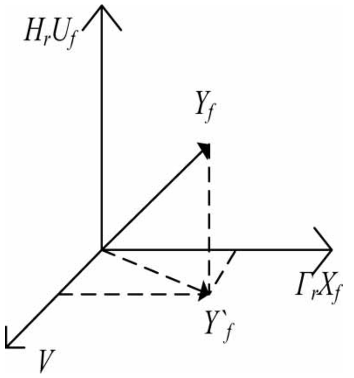

Fig. 5 Perpendicular projection of the future outputs perpendicular to the future inputs

$$\mathbf{U}_{\mathrm{f}}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}} = \mathbf{U}_{\mathrm{f}} - \mathbf{U}_{\mathrm{f}}\mathbf{U}_{\mathrm{f}}^{\mathrm{T}}(\mathbf{U}_{\mathrm{f}}\mathbf{U}_{\mathrm{f}}^{\mathrm{T}})^{-1}\mathbf{U}_{\mathrm{f}} = \mathbf{U}_{\mathrm{f}} - \mathbf{U}_{\mathrm{f}}\mathbf{I} = 0 \tag{13}$$

Postmultiplying both sides of equation (11.2) by the projection operator $\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}}$ yields

$$\begin{aligned} [\boldsymbol{Y}_{\mathrm{f}}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}}]_{r\times N} &= \Gamma_r\mathbf{X}_{\mathrm{f}}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}} + \mathbf{H}_r\mathbf{U}_{\mathrm{f}}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}} + \mathbf{V}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}} \\ &= [\Gamma_r\mathbf{X}_{\mathrm{f}}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}}]_{r\times N} + [\mathbf{V}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}}]_{r\times N} \end{aligned} \tag{14}$$

The input contribution has now been removed from the prediction equation.

2.3.2.2 Instrument variable to remove the noise contribution. It is now necessary to exclude the noise term $\mathbf{V}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}}$ of equation (14). This can be done by postmultiplying equation (14) with a suitable matrix $\boldsymbol{\Phi}$ that is not correlated to the noise matrix$\mathbf{V}$. Let matrix $\boldsymbol{\Phi}$ can be defined as follows

$$\boldsymbol{\Phi} = \begin{bmatrix}\varphi_s(t) & \varphi_s(t+1) & \cdots & \varphi_s(t+j-1)\end{bmatrix}_{s\times N} \tag{15}$$

where $\boldsymbol{\varphi}_s$ are vectors that are uncorrelated with the noise. The number of lines s of this matrix may have any value, but it has to be higher than the desired order n of the dynamic system. This variable is called an instrument variable [**10**, **12**] and it is used to reduce the noise term of equation (14) to zero. Next, postmultiply equation (15) by the transpose of $\boldsymbol{\Phi}$ and normalize with the sample number (by dividing the equation by N) in the data as follows

$$\begin{aligned} [\mathbf{G}]_{ro\times s} &\overset{\mathrm{def}}{=} \frac{1}{N}\boldsymbol{Y}_{\mathrm{f}}\boldsymbol{\Pi}^{\perp}_{\mathbf{U}^{\mathrm{T}}}\boldsymbol{\Phi}^{\mathrm{T}} = \frac{1}{N}\boldsymbol{\Gamma}_r\mathbf{X}_{\mathrm{f}}\boldsymbol{\Pi}^{\perp}_{\mathbf{U}^{\mathrm{T}}}\boldsymbol{\Phi}^{\mathrm{T}} \\ &\quad + \frac{1}{N}\mathbf{V}\boldsymbol{\Pi}^{\perp}_{\mathbf{U}^{\mathrm{T}}}\boldsymbol{\Phi}^{\mathrm{T}} \overset{\mathrm{def}}{=} [\boldsymbol{\Gamma}_r\mathbf{T}_N]_{ro\times s} + [\mathbf{F}_N]_{ro\times s} \end{aligned} \tag{16}$$

In equation (16), the subscript N implies that the value of $\mathbf{T}_N$ and $\mathbf{F}_N$ are approximated for a data record containing N data points. In order to cancel out the noise term without affecting the term dependant on the future states $\mathbf{T}_N$, the requirement for a proper instrument matrix $\boldsymbol{\Phi}$ is that it must be correlated to the future states $\mathbf{X}_\mathrm{f}$, but uncorrelated with the noise term. Mathematically, it can be expressed by the following equations

$$\lim_{N\to\infty} \mathbf{F}_N = \lim_{N\to\infty} \frac{1}{N}\mathbf{V}\Pi^{\perp}_{\mathbf{U}^\mathrm{T}}\boldsymbol{\Phi}^\mathrm{T} = 0 \tag{17}$$

$$\lim_{N\to\infty} \mathbf{T}_N = \lim_{N\to\infty} \frac{1}{N}\mathbf{X}_\mathrm{f}\Pi^{\perp}_{\mathbf{U}^\mathrm{T}}\boldsymbol{\Phi}^\mathrm{T} = \mathbf{T} \tag{18}$$

In equation (18), the parameter $\mathbf{T}$ is equivalent to the estimation of $\mathbf{T}_N$ with an infinite number of data points. Equation (17) implies that as the number of samples N goes to infinity, the noise matrix and the instrument matrix $\boldsymbol{\Phi}$ must cancel each other and $\mathbf{F}_N$ goes to zero. Equation (16) can therefore be summarized as

$$\mathbf{G} = \frac{1}{N}\boldsymbol{Y}_\mathrm{f}\Pi^{\perp}_{\mathbf{U}^\mathrm{T}}\boldsymbol{\Phi}^\mathrm{T} \simeq \boldsymbol{\Gamma}_r\mathbf{T} \tag{19}$$

The remaining step of the above demonstration is to find an appropriate instrument matrix $\boldsymbol{\Phi}$. As stated earlier, the first requirement for the instrument matrix $\boldsymbol{\Phi}$ is that it must be correlated with the future states, which are unknown at this point. Even though these states are unknown, their values are dependent on the past inputs and outputs, so these past inputs and outputs present possible choices. The second requirement is that the matrix $\boldsymbol{\Phi}$ must be uncorrelated with the noise. This is always the case when the system inputs are properly selected, because the noise can be seen as the error between the model and the data and, for a good model, this error is completely random. The past inputs and outputs are therefore a logical choices for the instrument matrix. In the current paper, the instrument matrix used is described in equation (20), which was taken from reference [**10**]

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{Y}_\mathrm{p} \\ \mathbf{U}_\mathrm{p} \end{bmatrix}_{s\times N}$$
$$= \begin{bmatrix} \boldsymbol{y}[t-1]_{o\times 1}\ \ \boldsymbol{y}[t]_{o\times 1}\ \ \ldots \boldsymbol{y}[t+j-2]_{o\times 1} \\ \ldots\ \ \ldots\ \ \ldots\ \ \ldots \\ \boldsymbol{y}[t-h]_{o\times 1}\boldsymbol{y}[t-h+1]_{o\times 1}\ \ \ldots \\ \boldsymbol{y}[t-h+j-1]_{o\times 1} \\ - - - - - - - \\ \boldsymbol{u}[t-1]_{m\times 1}\ \ \boldsymbol{u}[t]_{m\times 1}\ \ \ldots\ \ \boldsymbol{u}[t+j-2]_{m\times 1} \\ \ldots\ \ \ldots\ \ \ldots\ \ \ldots \\ \boldsymbol{u}[t-h]_{m\times 1}\ \ \boldsymbol{u}[t-h+1]_{m\times 1}\ \ \ldots \\ \boldsymbol{u}[t-h+j-1]_{m\times 1} \end{bmatrix}_{s\times N} \tag{20}$$

In this equation, the subscript p stands for past inputs and outputs and the parameter h is the number of past inputs and outputs used by the algorithm. Once a proper instrument matrix has been found, the output equation can be described with equation (19). Inserting equation (16) into equation (19) yields

$$\frac{1}{N}\boldsymbol{Y}_\mathrm{f}\Pi^{\perp}_{\mathbf{U}^\mathrm{T}}\boldsymbol{\Phi}^\mathrm{T} = \frac{1}{N}\boldsymbol{\Gamma}_r\mathbf{X}_\mathrm{f}\Pi^{\perp}_{\mathbf{U}^\mathrm{T}}\boldsymbol{\Phi}^\mathrm{T} \tag{21}$$

At this point, every term on the left-hand side of equation (21) is known. The next step will be to extract the observability matrix $\boldsymbol{\Gamma}_r$, which can be done by performing a singular value decomposition (SVD). More details on this procedure will be explained in section 3.

2.3.3 *Determination of the observability matrix from SVD*

Once matrix $\mathbf{G}$ is known from equation (19), it is possible to decompose it into three sub-matrices using a well-known linear algebra theorem called SVD. This theorem is explained in thorough detail in reference [**14**]. This decomposition is expressed as follows

$$\mathbf{G}_{ro\times s} = \mathbf{U}_{ro\times ro}\mathbf{S}_{ro\times s}\mathbf{V}^\mathrm{T}_{s\times s} \tag{22}$$

In equation (22), the matrix $\mathbf{S}$ is made up of the singular values of $\mathbf{G}$. These singular values are the positive square root of the eigenvalues of ($\mathbf{G}^\mathrm{T}\mathbf{G}$), and these eigenvalues are sorted in descending order from the first row to the last row of matrix $\mathbf{S}$. The matrices $\mathbf{U}$ and $\mathbf{V}$ are called singular vectors. They are the orthonormal eigenvectors of ($\mathbf{G}\mathbf{G}^\mathrm{T}$) and ($\mathbf{G}^\mathrm{T}\mathbf{G}$), respectively. The following demonstration will show how the new matrices defined by the SVD can be used to obtain an extended observability matrix $\boldsymbol{\Gamma}_r$ that correctly relates the inputs to the outputs. SVD provides a possible combination of matrices whose product gives the matrix $\mathbf{G}$. Of these two matrices, the first is a possible observability matrix and the second is a possible state vector. The SVD of equation (22) applies for a dynamic system of any order. If the desired order n of the dynamic system is known, the correct procedure separates the first n singular values and singular vectors of the system from the other singular values and vectors. In practice, when performing a SVD, only the system's significant singular values and their corresponding singular vectors must be kept. The number of singular values considered corresponds to the assumed order of the model. The singular values considered should represent the true dynamic of the system and the small singular values that are not taken into account should correspond to errors because of noise. Once the SVD is done, it is possible to obtain a proper estimate of the observability matrix $\boldsymbol{\Gamma}_r$. Notice that many different combinations of observability matrices and $\mathbf{T}$ matrices can lead to a set of parameter values that insure a proper match. According to Ljung [**10**], it follows that the value of the observability matrix may be expressed with the following equation

$$\boldsymbol{\Gamma}_r = \mathbf{U} \tag{23}$$

It is also possible to add weight functions to the matrix **G** obtained from equation (23) before performing SVD as follows

$$\mathbf{G}_{\text{Weight}} = \mathbf{W}_1\mathbf{G}\mathbf{W}_2 \tag{24}$$

The reason for adding weight matrices is to remove any residual error because of noise in the variable **G**. These weight matrices are made up of parameters that are uncorrelated with the projected noise matrix $\mathbf{F}_N$. In the absence of noise, adding a weight matrix has no effect on the identification results, but it improves the results when the identification is done on noisy data.

After a weight matrix is selected, the new observability matrix is found by use of the following equation

$$\boldsymbol{\Gamma}_r = \mathbf{W}_1^{-1}\mathbf{U} \tag{25}$$

Many authors have proposed expressions for different weight matrices, and they have an influence on the results of the identification. A good summary is available in reference [**12**], where different expressions for weight matrices are derived using the same mathematical approach. For original work on the different weight matrices, refer references [**15**] to [**18**]. In the current paper, the weight matrices defined by Larimore [**17**] were tried and gave good results. Since there is not much noise in the data related to this project, the algorithm is not very sensitive to the selected weight and it was not necessary to try other weight formulations to obtain good results. The weights as defined by Larimore are defined by use of the following equation

$$\mathbf{W}_1 = \left(\frac{1}{N}\boldsymbol{Y}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}}\boldsymbol{Y}^{\mathrm{T}}\right)^{-1/2} \quad ; \quad \mathbf{W}_2 = \left(\frac{1}{N}\boldsymbol{\Phi}\Pi^{\perp}_{\mathbf{U}^{\mathrm{T}}}\boldsymbol{\Phi}^{\mathrm{T}}\right)^{-1/2} \tag{26}$$

At this point the observability matrix $\boldsymbol{\Gamma}_r$ has been determined from equation (25). This observability matrix can now be used to obtain the value of the matrices [**A**, **B**, **C**, **D**]. The procedure is shown in the next section.

*2.3.4 Determination of the system matrices [**A**, **B**, **C**, **D**] from the observability matrix*

*2.3.4.1 Estimating **A** and **C**.* Once the observability matrix is known, it is quite easy to obtain estimates of the **A** and **C** matrices. Although equation (4) is referred to as it was defined at the beginning of the theory section, the estimate matrix **C** is obtained by taking the following terms of the observability matrix

$$\hat{C} = \boldsymbol{\Gamma}_{(1:o,1:n)} \tag{27}$$

where the hat '^' means that it is an estimate. Equation (27) simply means that we have to extract the first *o* lines and *n* columns of the observability matrix are to be extracted. The matrix **A** can be found from the observability matrix by solving the following equation

$$\boldsymbol{\Gamma}_{(o+1:ro,1:n)} = \boldsymbol{\Gamma}_{(1:o(r-1),1:n)}\mathbf{A} \tag{28}$$

In this equation, the left-hand side represents the observability matrix with the first submatrix $\hat{\mathbf{C}}$ removed and the right-hand side represents the observability matrix with the last submatrix $\mathbf{CA}^{r-1}$ removed. This is equivalent to the following equation

$$\begin{bmatrix} \mathbf{CA}_{o\times n} \\ \mathbf{CA}^2_{o\times n} \\ \cdots \\ \mathbf{CA}^{r-1}_{o\times n} \end{bmatrix}_{(r-1)o\times n} = \begin{bmatrix} \mathbf{C}_{o\times n} \\ \mathbf{CA}_{o\times n} \\ \cdots \\ \mathbf{CA}^{r-2}_{o\times n} \end{bmatrix}_{(r-1)o\times n} \mathbf{A}_{n\times n} \tag{29}$$

where the only unknown is the state matrix **A**. If the modified observability matrices $\boldsymbol{\Gamma}_{(o+1:ro,1:n)}$ and $\boldsymbol{\Gamma}_{(1:o(r-1),1:n)}$ of equation (29) were both square matrices, it would be easy to find $\hat{\mathbf{A}}_{n\times n}$ by simply premultiplying both sides of the equation by the inverse of $\boldsymbol{\Gamma}_{(1:o(r-1),1:n)}$. Since these matrices are not necessarily square, equation (29) can be solved by use of the following equation

$$\hat{\mathbf{A}}_{n\times n} = \left(\begin{bmatrix} \mathbf{C}_{o\times n} \\ \mathbf{CA}_{o\times n} \\ \cdots \\ \mathbf{CA}^{r-2}_{o\times n} \end{bmatrix}^{\dagger}\right)_{n\times(r-1)o} \begin{bmatrix} \mathbf{CA}_{o\times n} \\ \mathbf{CA}^2_{o\times n} \\ \cdots \\ \mathbf{CA}^{r-1}_{o\times n} \end{bmatrix}_{(r-1)o\times n} \tag{30}$$

In equation (30), the superscript '†' denotes the Moore–Penrose pseudo-inverse [**13**]. This is a more general type of inversion, which does not require the matrices to be square. The pseudo-inverse can be computed by SVD (refer references [**14**] and [**18**] for more details).

*2.3.4.2 Estimating **B** and **D**.* Once the system matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{C}}$ are known, it is possible to estimate the **B** and **D** matrices by use of a linear regression technique [**10**]. The discrete state-space model of equation (2) and (3) can be converted into a discrete transfer function by using the discrete operator *z* defined with the following equation

$$z\boldsymbol{x}(t) = \boldsymbol{x}(t + \Delta t)$$
$$z^{-1}\boldsymbol{x}(t) = \boldsymbol{x}(t - \Delta t) \tag{31}$$

which leads to the following transfer function

$$\hat{\boldsymbol{y}}(t|\mathbf{B}, \mathbf{D}) = \hat{\mathbf{C}}(z\mathbf{I} - \hat{\mathbf{A}})^{-1}\mathbf{B}\boldsymbol{u}(t) + \mathbf{D}\boldsymbol{u}(t) \tag{32}$$

where $\hat{\boldsymbol{y}}(t|\mathbf{B}, \mathbf{D})$ means the estimated output that depends on the values of the matrices **B** and **D**. A very efficient way to find the unknown parameters **B** and **D** of equation (32) is to use a linear regression method. The estimated output $\hat{\boldsymbol{y}}(t)$ may be expressed by the following equation

$$\hat{\boldsymbol{y}}(t)_{o\times 1} = \boldsymbol{\eta}(t)\hat{\theta} = \boldsymbol{\eta}(t)_{o\times(nm+om)}\begin{bmatrix}\mathrm{Vec}(\mathbf{B})\\ \mathrm{Vec}(\mathbf{D})\end{bmatrix}_{(nm+om)\times 1} \tag{33}$$

In equation (33), the matrix $\boldsymbol{\eta}(t)$ is composed of the past and presents inputs $\boldsymbol{u}(t)$. The single column vector $\hat{\boldsymbol{\theta}}$ represents all the estimated parameters to be found by the regression – these unknown parameters are all the elements of the matrices **B** and **D**. The operator 'Vec' builds a column vector from a matrix by stacking its columns on top of each other.

2.3.5 Selection of the mathematical model order

The aircraft model structure should be defined with enough parameters to obtain good results, but should be parsimonious because a model with too many parameters may overfit the data. The Akaike information criterion [**19**] was used to compute a modified cost function J that takes into account the number of parameters d_{M} used to define the model. This criterion is defined as follows

$$J = \frac{1 + d_{\mathrm{M}}}{1 - d_{\mathrm{M}}}\left\{\frac{1}{N}\sum_{i=1}^{N}[\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)]^2\right\} \tag{34}$$

where the cost function J decreases when the mean square error between the model and the data decreases, but increases with the number of model parameters d_{M}. The best model is the optimal compromise between a low mean square error and a model with a small number of parameters and will actually have the lowest possible cost function. In Table 1, the cost functions J are calculated using equation (34) for the second, third, and fourth-order models for all nine MISO models. The minimum value of J is found with the third model order, which is therefore chosen to be the optimal model order.

Table 1 Model order selection based on cost function J values

Model order	Order 2	Order 3	Order 4
Cost function J	6.2591×10^{-6}	2.2699×10^{-6}	3.9055×10^{-5}

3 RESULTS

This section of this paper is divided into the following three parts.

1. An example of the goodness of the match between the model output and the flight test data is shown graphically for the flight condition characterized by Mach number = 0.85 and an altitude of 5000 ft.
2. The criteria used to evaluate the model for all flight conditions are explained.
3. The results for every flight conditions in terms of these criteria are explained.

3.1 Results for the flight condition characterized by Mach number = 0.85 and altitude of 5000 ft

Figure 6 shows the nine structural deflections on different parts of the aircraft with respect to time. The results presented in Fig. 6 are given for a flight test condition at 5000 ft at a constant Mach number of 0.85. The model's estimation results are represented by full lines, and the stars represent the flight flutter tests data.

From Fig. 6, the estimated surface deflections follow the same dynamics as the surface deflection from flight flutter tests.

3.2 Criteria used to evaluate the results

Three methods are used to evaluate the model's accuracy: the correlation coefficient, the fit coefficient, and the robustness test. These methods are explained in this section.

3.2.1 Correlation coefficient

The concept of correlation coefficient may be illustrated by plotting the results from flight flutter test data versus the estimated (calculated) results expressed in terms of structural surfaces' deflections as shown in Fig. 7.

The correlation coefficient R is defined as a measure of the scatter in the graph shown in Fig. 7 between the output from flight flutter test data and the calculated

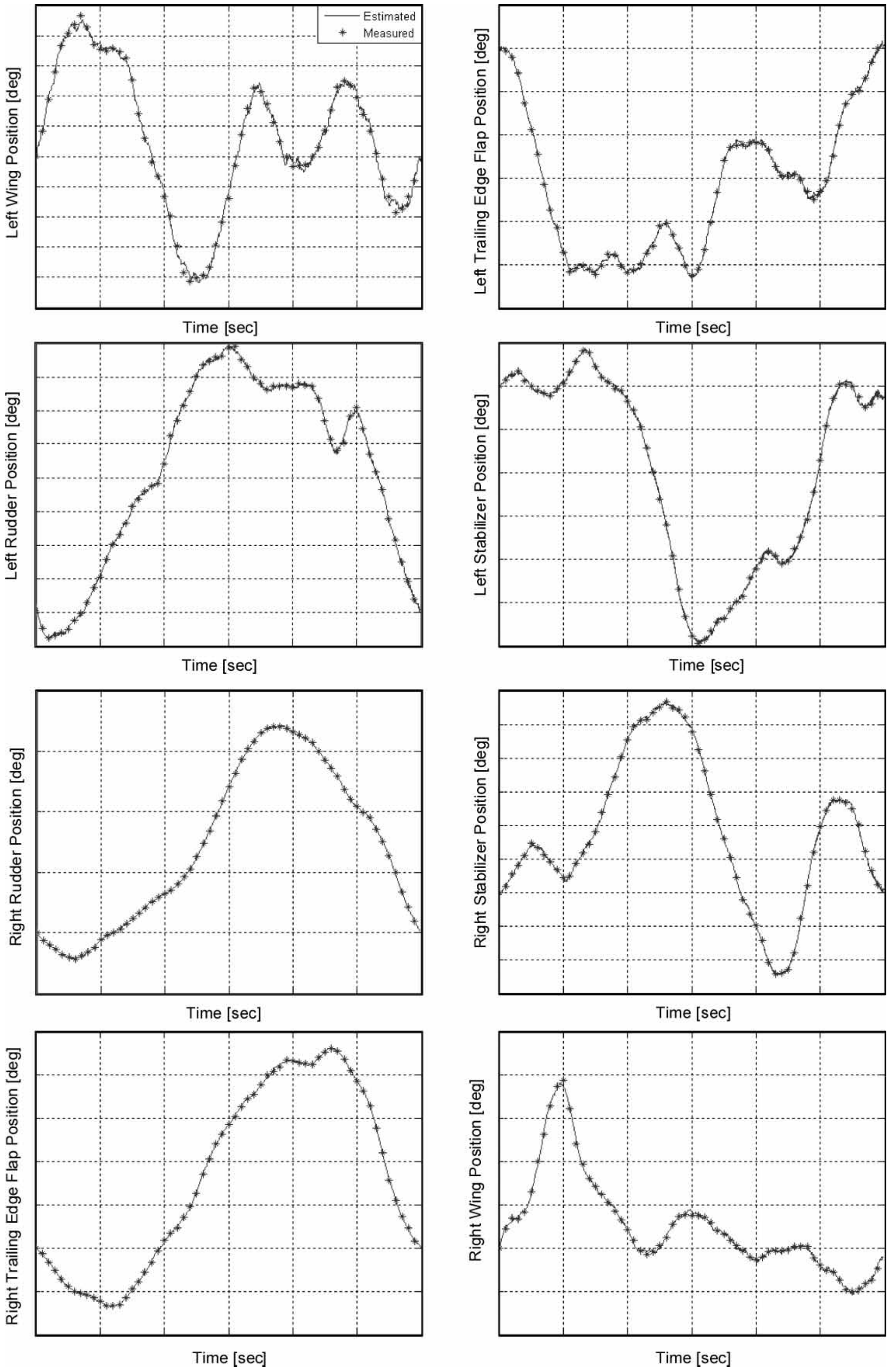

Fig. 6 State-space model estimation of structural surface deflections (full line) and their measurements (or flight tests data) with respect to time

(estimated) output. Mathematically, its expression is given by the following equation

$$R = \frac{\mathrm{Cov}(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\sqrt{\mathrm{Var}(\boldsymbol{y})\mathrm{Var}(\hat{\boldsymbol{y}})}} \tag{35}$$

where Cov is the covariance, Var is the variance, $\boldsymbol{y}$ is the output from flight flutter test data, and $\hat{\boldsymbol{y}}$ is the estimated output. The value of the correlation coefficient R is situated between -1 and 1. The correlation coefficient $R = 1$ denotes perfect linear dependency

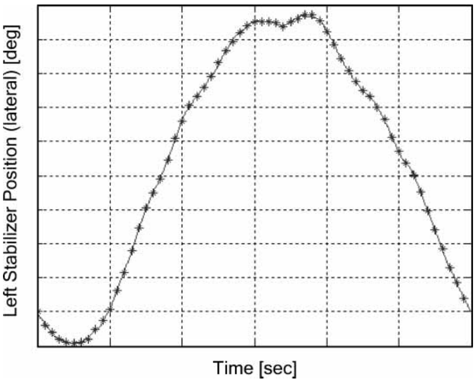

Fig. 6 (Continued)

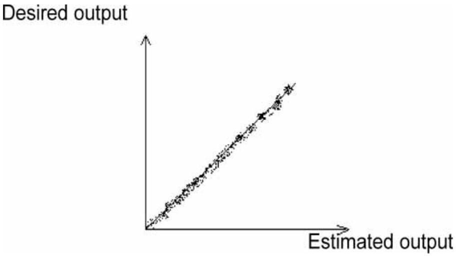

Fig. 7 Visual interpretations of the estimated structural surfaces' deflection outputs versus the output from flight flutter test data

(no scatter) between the outputs from flight flutter test data with respect to the calculated or estimated outputs. A correlation coefficient equal to minus one ($R = -1$) denotes inverse linear dependency between the estimated output and the output from flight flutter test data. A correlation coefficient of zero ($R = 0$) denotes the linear independency between the estimated outputs and the output from flight flutter tests. The correlation coefficient computes the goodness of the model in a statistical sense, but provides little information about the model error. More information can be obtained by the second method: the fit coefficient.

3.2.2 *Fit coefficient*

The fit coefficient is defined as 100 per cent multiplied by the ratio between the L_2-norm of the error between the data and the model over the L_2-norm of the error between the data and its mean value. The fit coefficient is expressed by equation (36)

$$\text{FIT} = 100\left(1 - \frac{\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|}{\|\boldsymbol{y} - \text{mean}(\boldsymbol{y})\|}\right) \tag{36}$$

where the term in ($\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|$) is the L_2-norm of the error. The L_2-norm is defined with equation (37)

$$\|\boldsymbol{y} - \hat{\boldsymbol{y}}\| = \frac{1}{2}\sqrt{\sum_{1}^{n}(y - \hat{y})^2} \tag{37}$$

Equation (36) therefore becomes

$$\text{FIT} = 100\left\{1 - \sqrt{\frac{\sum_1^n(\boldsymbol{y} - \hat{\boldsymbol{y}})^2}{\sum_1^n[\boldsymbol{y} - \text{mean}(\boldsymbol{y})]^2}}\right\} \tag{38}$$

In equation (38), the term under the square root represents the ratio between two residuals: the residual between the model and the data and the residual between the data and their mean value. Intuitively, the fit coefficient explains the percentage of data variation that is explained by the model. The main advantage of using the fit coefficient is that it takes into account how the data varies about its mean in order to evaluate the model quality. For example, even if a model has an output very close to the data output, it will have a poor fit coefficient if there are many more small oscillations in the data than in the model.

3.2.3 *Robustness test*

In order to test the estimated model's robustness with flight test data, resampled signals that use one point over one hundred (100) are considered. The signals are further resampled, because the same sampling rate as that used in the model implementation ($\mathbf{T}_s = 0.01$) is to be used (Fig. 8).

The schematic shown in Fig. 8 is equivalent to a procedure that makes perturbations on the model inputs and outputs in order to check its robustness. The reconstructed inputs are further used in the initial model. This is equivalent to adding many small perturbations to the inputs signals, in order to measure the sensitivity of the model to these perturbations. In Fig. 9

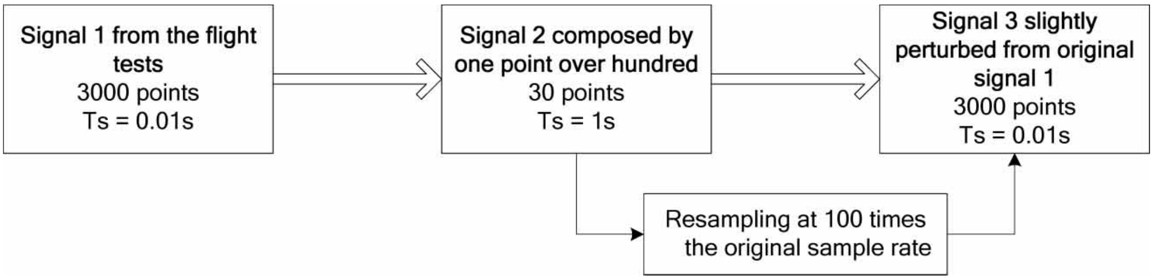
Signal 1 from the flight tests
3000 points
Ts = 0.01s
Signal 2 composed by one point over hundred
30 points
Ts = 1s
Signal 3 slightly perturbed from original signal 1
3000 points
Ts = 0.01s
Resampling at 100 times the original sample rate

Fig. 8 Robustness test methodology

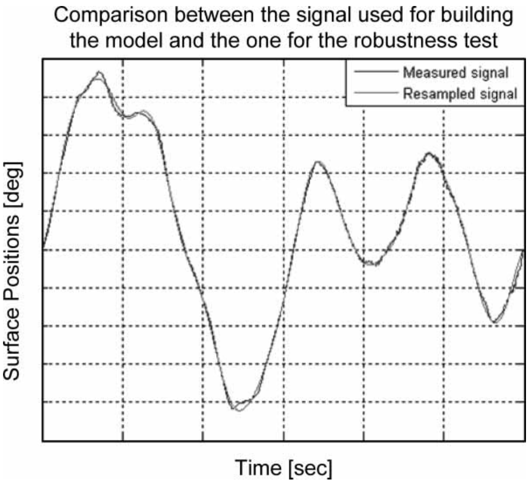

Fig. 9 Initial flight flutter test data versus resampled signals

the initial signals from the flight tests are very close to the resampled signals.

If the model is robust, the model output from the resampled input should be similar to the model output when using the initial input.

3.3 Results for all flight flutter test conditions

In order to summarize the results obtained by the identification and the robustness tests, the average correlation and fit coefficients for all flight conditions were plotted on the same figure, and for each surface deflection in the upper and lower parts of Fig. 10. In this figure, the full bars represent the results from the identification and the dashed bars represent the results of the robustness test. The black lines represent the standard deviation of each coefficient for all flight conditions.

First, it is observed that the fit coefficients are very good for the identification part (model construction), as the average fit coefficient is found to be higher than 97 per cent for each surface, which shows that the calculated model validates the real aircraft system (from flight tests).

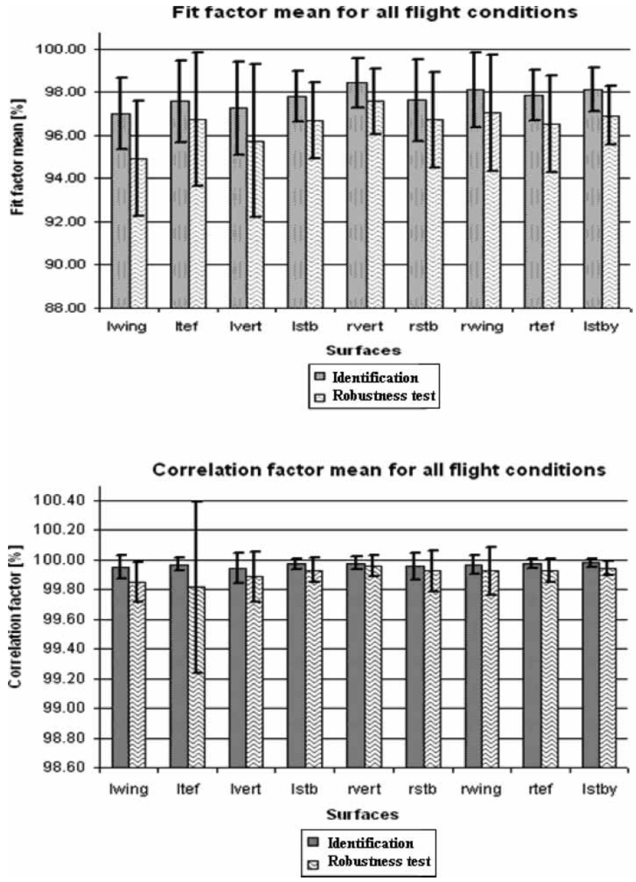

Fig. 10 Correlation and fit factors mean and standard deviation for all flight conditions

It can also be noted that for the robustness tests, fit and correlation coefficients of almost the same magnitude as those obtained in the identification tests were obtained. The differences between the two sets of results are 1–2 per cent. It can therefore to concluded that the estimated model is robust.

4 CONCLUSIONS

Nine third-order linear state-space MISO models were used in this study to estimate the structural surface positions given by the F/A-18 differential ailerons control inputs for flight conditions characterized by different Mach numbers and altitudes. Nineteen flight tests with different combinations of Mach numbers and altitudes were considered in this study. The subspace method was used for the model identification from flight flutter tests. Two methods were used to evaluate this model's performance: the correlation coefficients definition and the fit coefficients definition.

The correlation coefficients are close to 100 per cent and the fit coefficients also have high values, with the worse values at 89 per cent. Therefore, the estimated linear model fits the flight flutter tests data very well.

The advantage of the subspace identification method is its small computation time and also the estimation of a very good model from the knowledge of the flight tests' inputs and outputs without *a priori* knowledge about the model dynamics. The estimated model was found to be robust by the resampling technique. From the obtained results it can be concluded that the subspace method is very convenient and efficient for model identification from flight flutter tests.

ACKNOWLEDGEMENTS

The authors would like to thank Mr Marty Brenner at NASA Dryden Research Flight Center for his collaboration. Financial support for the work related to this paper was given by the Natural Sciences and Engineering Research Council of Canada NSERC and by the Ministère du Développement économique, de l'innovation et de l'exportation MDEII.

REFERENCES

1 **Voracek, D., Pendleton, E., Reichenback, E., Griffin, K.,** and **Welch, L.** The active aeroelastic wing phase I flight research through January 2003. NASA/TM-2003-210741, 2003.

2 **Lind, R.** and **Brenner, M.** *Robust aeroservoelastic stability analysis: flight test applications* 1999 (Springer-Verlag, London).

3 **Sung Won, K., Tsai, H. M., Sadeghi, M.,** and **Liu, F.** Nonlinear impulse methods for aeroelastic simulations. In 23rd AIAA Applied Aerodynamics Conference, Toronto, Canada, 6–9 June 2005, pp. 1–19.

4 **Kukreja, S. L.** and **Brenner, M.** Nonlinear aeroelastic system identification with application to experimental data. *AIAA, J. Guid. Control Dyn.*, 2006, **29**(2), 374–381.

5 **Silva, W. A., Vartio, E., Shimko, A., Segundo, E. I., Kvaternik, R. G., Eure, K. W.,** and **Scott, R.** Development of aeroservoelastic analytical models and gust load alleviation control laws of a sensorcraft wind-tunnel model using measured data. 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Newport, USA, 4–6 May 2006, paper AIAA-2006-1935.

6 **Le Garrec, C., Humbert, M., Bucharles, A.,** and **Vacher, P.** In flight aeroelastic model identification and tuning of a flight control system on a large civil aircraft. CEAS International Forum on Aeroelasticity and Structural Dynamics, Madrid, Spain, 5–7 June 2001.

7 **Galvao, R. K. H., Hadjiloucas, S., Becerra, V. M.,** and **Bowen, J. W.** Subspace system identification framework for the analysis of multimoded propagation of THz-transient signals. *Meas. Sci. Technol.*, 2005, **16**, 1037–1053.

8 **Brenner, M.** and **Feron, E.** Wavelet analysis of F/A-18 aeroelastic and aeroservoelastic flight test data. 38th AIAA/ASME/ASCE/AHS/ASC Structure, Structural Dynamics and Material Conference Exhibit, Kissimmee, FL, USA, 7–10 April 1997, paper AIAA-97-1216.

9 **Schroeder, M. R.** Synthesis of low-peak factor signals and binary sequences with low autocorrelation. *IEEE Trans. Inf. Theory*, 1970, **16**, 85–89.

10 **Ljung, L.** *System identification theory for the user*, 2nd edition, 1999, pp. 340–351 (Prentice Hall, Upper Saddle River, NJ).

11 **Lennart, L.** *System identification toolbox for use with Matlab®. User guide version 6*, 2006 (The Mathworks Inc.).

12 **Viberg, M., Wahlberg, B.,** and **Ottersen, B.** Analysis of state space system identification methods based on instrumental variables and subspace fitting. *Automatica*, 1997, **33**(9), 1603–1616.

13 **Viberg, M.** Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 1995, **31**(12), 1835–1852.

14 **Patel, R. V., Laub, A. J.,** and **Van Dooren, P. M.** *Numerical linear algebra techniques for system and control*, 1993, p. 724 (IEEE Press, City College of New York).

15 **Verhaegen, M.** Identification of the deterministic part of MIMO state space models, given in innovation form from input-output data. *Automatica*, 1994, **30**(1), 61–74.

16 **Van Overschee, P.** and **DeMoor, B.** N4SID: subspace algorithm for the identification of combined deterministic-stochastic systems. *Automatica*, 1994, **30**(1), 75–93.

17 **Larimore, W. E.** Canonical variate analysis in identification, filtering, and adaptive control. In Proceedings of 29th IEEE Conference on Decision and Control, Honolulu, Hawaii, 1990, pp. 596–604.

18 **Klema, V. C.** and **Laub, A. J.** The singular value decomposition: its computation and some applications. *IEEE Trans. Autom. Control*, 1980, **25**(2), 164–176.

19 **Akaike, H.** Fitting autoregressive models for prediction. *Ann. Inst. Stat. Math.*, 1969, **21**, 243–347.

APPENDIX

Notation

$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$	matrices describing the discrete state-space model
A	excitation amplitude
d_{M}	number of component parameters in the model
$\mathbf{F}$	projected noise matrix multiplied by the instrument

$\mathbf{G}$	projected output matrix multiplied by the instrument
$\mathbf{H}$	impulse response matrix
$\mathbf{I}$	dentity matrix
j	dummy index
J	corrected cost function
lwing, rwing	left and right wing position
lail, rail	left and right aileron position
ltef, rtef	left and right trailing edge flap position
lvert, rvert	left and right rudder position
lstb, rstb	left and right stabilizer position
lstby	left stabilizer lateral position
N	length of the data record
R	correlation coefficient
$\mathbf{S}$	matrix of singular values
t, Δt	time, time step
$\mathbf{T}$	projected state matrix multiplied by the instrument
$\boldsymbol{u}$	system input vector
$\mathbf{U}$	matrix of past or future inputs or matrix of singular vector
$\boldsymbol{v}$	perturbation vector on the system outputs
$\mathbf{V}$	future noise effect matrix or matrix of singular vector
Vec	operation that organizes the parameters of a matrix into a column vector
$\boldsymbol{w}$	perturbation vector on the system states
$\mathbf{W}$	weight matrix
$\boldsymbol{x}$	system states vector
$\mathbf{X}$	matrix of past or future states
$\boldsymbol{y}$	output vector
z	discrete time shift operator
$\boldsymbol{\Gamma}$	observability matrix
$\boldsymbol{\Delta}$	controlability matrix
$\boldsymbol{\eta}$	matrix of past output combinations used in the linear regression
θ	unknown parameters used in the linear regression
Π	projection operator
ϕ	phase
φ	columns of the instrument matrix
$\boldsymbol{\Phi}$	instrument matrix
$\perp$	perpendicular projection
$\wedge$	estimated value
∞	infinity

Subscripts

f	future
h	number of past inputs used in the instrument matrix
k	kth time step prediction of noise effect (line k) on $\mathbf{V}$ matrix or kth frequency
m	number of inputs
n	number of states or order of the system
N	length of data vector or approximation with a finite amount of data points
o	number of outputs
p	past
r	forward prediction horizon
s	number of lines of the instrument matrix
Weight	$\mathbf{G}$ matrix with weight functions added

Superscripts

T	transpose
$\dagger$	Moore–Penrose pseudo-inverse