

# Flight dynamics helicopter model validation based on flight test data

A Hiliuta and R M Botez\*

Department of Automated Production Engineering, Ecole de Technologie Supérieure, Montréal, Québec, Canada

*The manuscript was received on 29 January 2007 and was accepted after revision for publication on 11 June 2007.*

DOI: 10.1243/09544100JAERO205

**Abstract:** In the current paper, a new technique for a helicopter model estimation based on flight test data is presented. The state space helicopter models, for different flight conditions, are computed. The method is automatic and is applied when one knows the motion parameters types  $x$ ,  $y$ , and  $u$  of a real helicopter and in the absence of any seed matrices. The method is highly convergent, and very good results are obtained for all flight cases. In the introductory part, a bibliographic research is carried on, and the necessary data collection is presented. In the next step, the preliminary estimation of seed matrices, for state-space representation is performed using a non-linear model and small perturbation theory. The last paragraph deals with the seed optimization with respect to different flight conditions. In the end, the model validation and the conclusion on possibilities to further develop and use the present methodology is investigated.

**Keywords:** helicopters, flight dynamics, flight tests, parameter estimation

## 1 INTRODUCTION

Aircraft flight model development is not an easy task and requires a huge number of data collected during the flight tests. The reason is that the aircraft is a complex aerodynamic vehicle in a complex environment. In such a model, one has to capture the aircraft behaviour within the atmosphere.

The main parameters of standard atmosphere, which have influence on the aircraft behaviour, are the total and dynamic pressure, the air density, and the air temperature. All these parameters vary as function of altitude (air pressure, temperature, and density) and speed (dynamic pressure). Therefore, changes in the motion parameters, such as altitude and speed, have direct impact on the forces and moments that act on the aircraft. A complete aircraft model has to capture all these changes. The dependencies between these parameters are very complex and lead to non-linear models.

In order to find the non-linear dependencies between the above mentioned parameters, one needs to estimate certain parameters fixed values and the slopes around these fix values. The fixed points are trim points, while the slopes of the curves in those points are estimated by applying small perturbations around them. This is the reason why the method is known as small perturbations theory. The perturbations from the trim point are induced by the pilot by use of the aircraft controls, one at the time. In order to capture the aircraft behaviour, a set of important parameters is recorded which is divided in three main categories: inputs  $u$ , outputs  $y$ , and states  $x$ .

The first category to be recorded is the set of control input parameters  $u$  which are designed to excite a useful frequency spectrum. Control input parameters types vary with the estimation methods used (in the time or in the frequency domain). The control inputs used in the time domain estimation methods are non-periodic rectangular pulses, with time widths of 3, 2, 1, and 1 s, and for this reason are defined as 3211 inputs [1]. In our application, the four controls of the helicopter (collective, longitudinal cyclic, lateral cyclic, and pedals) are used. However, applying a 3-s lateral control from trim condition was found to be too long for the aircraft flight limits attendance. For this reason, it was chosen to apply a 2-s control, followed

\*Corresponding author: Department of Automated Production Engineering, Ecole de Technologies Supérieure, 1100 Notre Dame West, Montréal, Québec, Canada H3C 1K3. email: ruxandra@gpa.etsmtl.ca

by a 3-s control. In this case, the aircraft was already deflected to one side by the 2-s control, while part of the 3-s control was used to return it into the original position. For this reason, the inputs were called 2311, instead of 3211.

The control inputs used in the frequency domain estimation methods are chirp or Schroeder signals. The chirp signals are also known as frequency sweep inputs and they start at very low frequency and increase to a predetermined maximum frequency [2]. Usually, the variation of frequency is linear,  $f(t) = f_0 t$ . The input signal can be defined as  $\sin \varphi(t)$ , when the chirp is added on trimmed input of the system, or  $A_0 \cos \varphi(t)$ , where  $A_0$  is the trim of the considered input. The phase  $\varphi(t)$  is the integral of the frequency and is given by

$$\varphi(t) = \int_0^t f_0 \tau \, d\tau = f_0 \frac{t^2}{2} \quad (1)$$

The Schroeder inputs are defined as a sum of components having different frequencies [3]

$$u(t) = \sum_{n=n_1}^{n=n_2} A_0 \sin(2\pi n f_0 t + \theta_n) \quad (2)$$

where  $f_0$  is the fundamental frequency,  $n_1$  and  $n_2$  impose the minimum and maximum frequency, and  $\theta_n = \pm\pi(n(n-1)/n_2 - n_1 + 1)$ . The both frequency sweep and Schroeder signal generate flat spectra to be applied as inputs. Moreover, Schroeder signal minimizes its peak-to-peak amplitude, very useful when an aircraft excitation is referred.

The interested reader can find details on the non-linear analysis and manoeuvre design for dynamic systems identification in specialized references [4].

The second category to be recorded is the set of output parameters  $\mathbf{y}$  which are the aircraft forces and moments. For the main application, the derived parameters related to the six-degrees of freedom (DOF) forces and moments are the linear and angular accelerations denoted with  $A_x, A_y, A_z, E_x, E_y, E_z$ .

The last category of parameters to be recorded is the set of state parameters  $\mathbf{x}$  which describe the state in which the aircraft is at one moment in time. In a flight dynamics model, one is interested in the aircraft motion, so that the state parameters are the motion parameters: linear and angular velocities, positions, and attitudes. For the main application, the linear body-fixed velocities, angular body-fixed velocities, the altitude, and the attitude Euler angles are recorded.

The aircraft linear model is described by a state-space linear system of equations

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{aligned} \quad (3)$$

where  $\mathbf{u}$  is the input vector,  $\mathbf{x}$  the state vector,  $\mathbf{y}$  the output vector, and  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  constant coefficients matrices which contain the linear dependencies between the parameters. The set of  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  matrices is calculated and generated for each helicopter flight condition. The most common methodology used for the calculation of  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  matrices involves a parameter estimation algorithm, which uses values of seed matrices for a close type aircraft. The algorithm changes  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  matrices elements until the outputs of simulated aircraft are similar to its recorded outputs. The chances for the algorithm to converge are much higher if the seed matrices are provided for the same aircraft in other flight condition or for another close aircraft. It is recommended to obtain the same values for the seed matrices to be used in the parameters estimation for all flight conditions, but this solution might be troublesome and time consuming.

In equation (3), the inputs  $\mathbf{u}$  are applied to the aircraft simulation model and the outputs  $\mathbf{y}$  are visualized. It is obvious that, in order to reach this goal, one needs to estimate the values of  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  matrices. Therefore, the parameter estimation problem consists in the calculation of the  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  matrices when  $\mathbf{u}$ ,  $\mathbf{x}$ , and  $\mathbf{y}$  are known.

The model generation method proposed in this paper has the following steps.

1. A trial-and-error technique (in our case an adaptive network fuzzy inference system – ANFIS) is used to map the measured inputs and outputs. The model obtained is an open loop non-linear model, using both  $\mathbf{x}$  and  $\mathbf{u}$  as inputs and  $\mathbf{y}$  as outputs.
2. The priority generated model is used to close the simulation loop by internally computing the states  $\mathbf{x}$  using six-DOF algorithm.
3. The trim point of the closed loop model is validated with respect to the flight test trim point (aircraft trim for the considered flight condition). A linear equivalent model (state-space) is obtained by use of small perturbations around the trim point and the Matlab<sup>®</sup> function 'dlinmod'.
4. The linear model is improved using an optimization technique which uses as starting point, the state-space matrices obtained at step 3, and an objective function to minimize the error between the model outputs and the recorded flight test data.

## 2 OPEN LOOP NON-LINEAR FUZZY MODEL

The goal of the final model is to generate the forces and moments as functions of state and input parameters. In fact, only the second equation from the set (3) is used. The state parameters are found by integrating the linear and angular accelerations using six-DOF

equations of motion, equations (7) and (9). For each flight condition, there is a unique set of  $\mathbf{C}$  and  $\mathbf{D}$  matrices that reproduces the  $\mathbf{y}$  vector when  $\mathbf{x}$  and  $\mathbf{u}$  vectors are the same as the recorded ones. These matrices will be obtained later through an optimization procedure. The closed loop model, in which the state parameters  $\mathbf{x}$  are obtained by integration from linear and angular accelerations, and fed back to compute these accelerations using the second equation (3), does not work if  $\mathbf{C}$ , and  $\mathbf{D}$  matrices are not correct. Due to this fact, the optimization procedure cannot start with randomly chosen seed matrices. Parameter estimation algorithms such as maximum likelihood estimation (MLE) algorithm use the starting guess seed matrices already calculated from a previous aircraft in order to compute the state-space matrices.

In this approach, a fuzzy model is used to find initial matrices, which are used as seed matrices in the future helicopter model development. The fuzzy model is generated by use of flight test data, and is composed of six fuzzy sub-models, three for the linear accelerations and three for the angular accelerations. The fuzzy models are multi-input-single-output Sugeno type systems [5]. The fuzzy subsystems are used to create an open loop simulation in which the linear and angular accelerations are generated by use of measured controls and aircraft motion state parameters. In this case, the inputs from flight tests to be considered are: the controls, linear velocities, angular velocities, and Euler angles. The outputs are the linear and angular accelerations. A short description of these fuzzy models is given in the following paragraphs.

In the classical set theory, the membership of an element to a set takes only two values: 0 and 1 (the element can belong or not to a set). In the fuzzy set theory, the membership of an element to a set has an infinity number of values, between 0 and 1 (the element belongs in a higher or in a lower measure to a set). The range of an input is divided, during the system design, over several fuzzy sets. Through a process called fuzzifying of the input, the membership of an input value to each fuzzy set is calculated. Logical rules, between the inputs, give weighting coefficients, which are called firing strengths. These coefficients are used to compute the total outputs  $\mathbf{y}$ , corresponding to the actual values of the inputs

$$\mathbf{y} = \frac{\sum w_i z_i}{\sum w_i} \quad (4)$$

where  $w_i$  are the weighting coefficients based on rules aggregation from activated membership functions, and  $z_i$  are, in the case of Sugeno system, a linear combination of the inputs  $u_j$

$$z_i = \sum a_{ij} u_j + b_i \quad (5)$$

The simplest Sugeno system is called zero-order, because of the fact that  $a_{ij}$  coefficients are zero and therefore,  $z_i$  are constants.

In order to build a fuzzy system, there is the need to find the correct number and corresponding shapes of the membership functions for each input  $u_j$ , and the constant values associated to each rule ( $z_i$ ), so that the output will best fit the measured data. Due to the large number of modifiable parameters, the algorithm to find the final values is iterative. The modifications in these parameters values are called training of the fuzzy system, and are achieved by use of a trial-and-error method called backpropagation algorithm [6]. These parameters are modified one after another and the training cycles, called epochs, end when all parameters are modified.

Following several training epochs, the output error decreases to an acceptable level. The fitting of the output parameters does not have to be very good in this phase of the model development, and for this reason, the training epochs could be limited to ten. This shows that after ten training epochs there is not significant improvement in fitting. The recreated outputs  $\mathbf{y}$ , obtained in a simulated environment by using as inputs the controls and the motion state parameters, are presented in Fig. 1.

The differences between the measured outputs (three linear accelerations  $A_x, A_y$ , and  $A_z$  and three angular accelerations  $E_x, E_y$ , and  $E_z$ ) and the simulated model outputs are very small, and for this reason, they seem superposed in Fig. 1. For confidentiality reasons, numerical values of results are not shown here and are normalized. The same reasons apply to all the figures showing time histories of measured flight parameters.

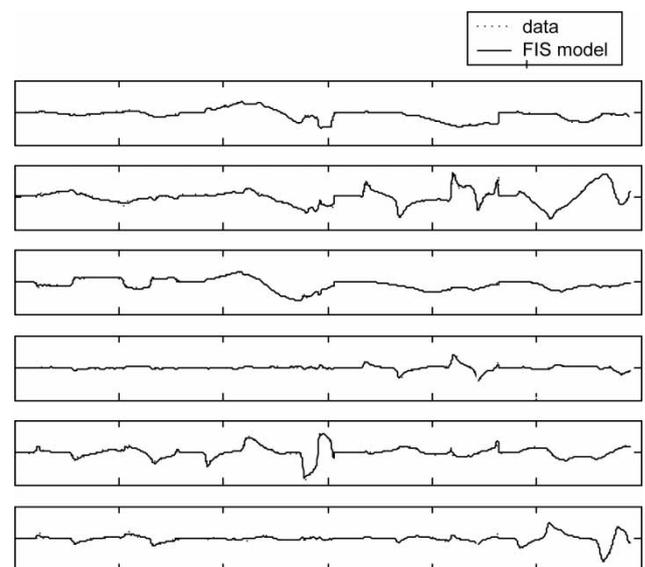


Fig. 1 Generated parameters using fuzzy system

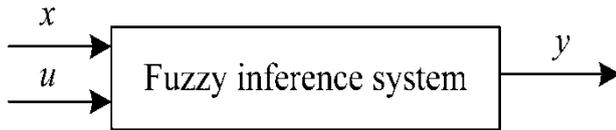


Fig. 2 Open loop non-linear model

The diagram of the open loop non-linear fuzzy inference system is shown in Fig. 2. The MATLAB fuzzy logic toolbox has been used to generate and train the fuzzy inference system.

### 3 CLOSED LOOP MODEL

The generated fuzzy system described in the previous section is an open loop system, where the inputs are the controls  $u$  and state parameters  $x$  (speeds and attitude angles). The state parameters  $x$  can be calculated by use of a six-DOF rigid body motion equations. These equations allow us to obtain the rigid body motion parameters (speeds and attitude angles) when the forces and moments acting on the body (which are expressed here in terms of the derived quantities linear and angular accelerations) are known. The closed loop diagram is presented in Fig. 3. Then, the measured state parameters  $x$  are replaced with the ones computed by this model.

Although the generated outputs in Fig. 1 are extremely close to the measured ones, the measurement errors values in state parameters  $x$  prevent the model to properly perform its task in closed loop. For the illustration of measurement error values, a comparison between the linear velocities  $u_b$ ,  $v_b$ , and  $w_b$  used for system generation ( $x$  in Fig. 2) and the ones used in closing the loop ( $x$  in Fig. 3) is given in Fig. 4.

The linear velocities in body axes  $u_b$ ,  $v_b$ , and  $w_b$  in Fig. 2 are calculated from true air speed (TAS), angle of attack ( $\alpha$ ) and sideslip angle ( $\beta$ ) measurements with following equations

$$\begin{aligned} u_b &= \text{TAS} \cos \alpha \cos \beta \\ v_b &= \text{TAS} \sin \beta \\ w_b &= \text{TAS} \sin \alpha \cos \beta \end{aligned} \quad (6)$$

In order to close the loop, the same quantities  $u_b$ ,  $v_b$ , and  $w_b$  in Fig. 3 are calculated by integrating linear

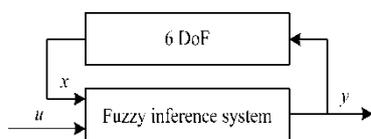


Fig. 3 Closed loop simulation model

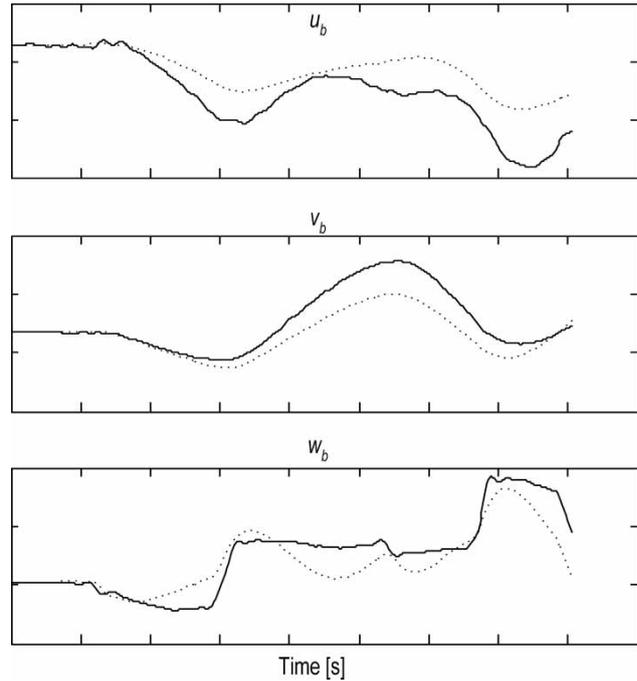


Fig. 4 Differences between body-axes linear velocities computed with equation (6) – solid, and equation (7) – dotted

accelerations in the six-DOF block as follows

$$\begin{aligned} \dot{u}_b &= A_x - qw_b + rv_b \\ \dot{v}_b &= A_y - ru_b + pw_b \\ \dot{w}_b &= A_z - pv_b + qu_b \end{aligned} \quad (7)$$

Therefore, small differences in measured accelerations at trim lead to cumulative measurement integration errors in  $u_b$ ,  $v_b$ , and  $w_b$  calculated with equations (6) and (7) as shown in Fig. 4.

Due to differences in state parameters  $x$  calculations above mentioned, in the closed loop conditions, the outputs are also modified, and a new adjustment of fuzzy system parameters would be necessary. There is no way to modify these parameters for a closed loop system. However, the block diagram in Fig. 3 can be used to obtain a linear state-space model. The used algorithm is based on small perturbation theory, estimating the sensitivities of each output with respect to each state and input, and is implemented and ready to use in Matlab<sup>®</sup>.

### 4 MODEL LINEARIZATION

The fuzzy system is linearized in a closed loop configuration by use of small perturbations theory around a trim point and, therefore, an equivalent state space

model is built. The validation of the closed loop helicopter model can be studied by comparison of the trim values of inputs, outputs and states parameters, obtained by model simulation versus real helicopter flight measurements. The differences between these steady state values are given in Table 1.

The equivalent linear helicopter model can be calculated at the trim point and the **A**, **B**, **C**, and **D** matrices are obtained by use of small perturbations theory. The linear model replaces both the aircraft model (the fuzzy subsystems) and the six-DOF block. The states can be computed by use of both equation (3) in the linear model, and of equation (7) in the non-linear model. By keeping the six-DOF block, only the second equation (3) is used which calculates the outputs **y** (linear and angular accelerations  $A_x, A_y, A_z, E_x, E_y,$  and  $E_z$ ) as functions of inputs **u** and states parameters **x**. In fact, the aircraft model is described by this second equation (3). The model adjustment can further be performed by optimization of **C** and **D** matrices. The response of the equivalent linear system, before optimization, is shown in Fig. 5. The integration of

states **x** is done by use of non-linear equation (7) for linear body-axis velocities  $u_b, v_b, w_b,$  and of non-linear equation (8) for Euler angles  $\phi, \theta,$  and  $\psi$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{8}$$

The angular velocities  $p, q, r$  are calculated from angular accelerations  $E_x, E_y,$  and  $E_z$  with

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix} \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix}^{-1} \times \begin{pmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{pmatrix} \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{9}$$

**Table 1** Steady-state differences (model – measured)

Steady-state differences	Values
$U_b$ (ft/s)	0.295 38
$V_b$ (ft/s)	0.295 38
$W_b$ (ft/s)	-0.018 051
$P$ (rad/s)	0.017 785
$Q$ (rad/s)	-0.001 099 2
$R$ (rad/s)	0.000 577 3

The aircraft model and the six-DOF block are depicted in Fig. 6.

The model response shown in Fig. 5 is not quite the same as the one from flight tests. Moreover, only one portion of the record is simulated, the model response diverging for longer simulation time. Therefore, the next step is necessary and mandatory in order to improve the model.

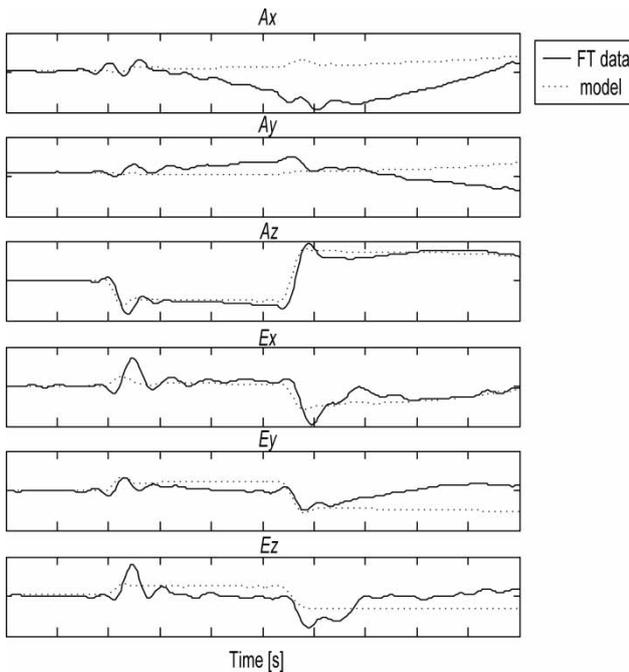
The **C** and **D** matrices are the starting point of future step: the optimization. The model adjustment, impossible for closed loop containing the fuzzy subsystem, can now be performed by optimizing **C** and **D** matrices.

**5 THE OPTIMIZATION PROCEDURE**

The optimization procedure is used to minimize the error between flight test data and the response of the model to the same input controls. The optimization goal is a multi-dimensional unconstrained non-linear minimization [7] of an objective function, especially designed for the application. This objective function minimizes the error between the model outputs **y** and the same measured outputs (linear and angular velocities  $A_x, A_y, A_z, E_x, E_y,$  and  $E_z$ ). The objective function employed to minimize errors is

$$J = \sum_{k=1}^{N_R} \left[ \sum_{i=1}^{N_0} \left( \sum_{j=1}^N |y_{meas}(i) - y_{model}(i)| \right) \right] \tag{10}$$

where  $y_{meas} = [A_x A_y A_z E_x E_y E_z]_{meas}$  are the measured quantities during flight test,  $y_{model} = [A_x A_y A_z E_x E_y E_z]_{model}$  are the outputs of the simulated model,



**Fig. 5** Equivalent linear system response

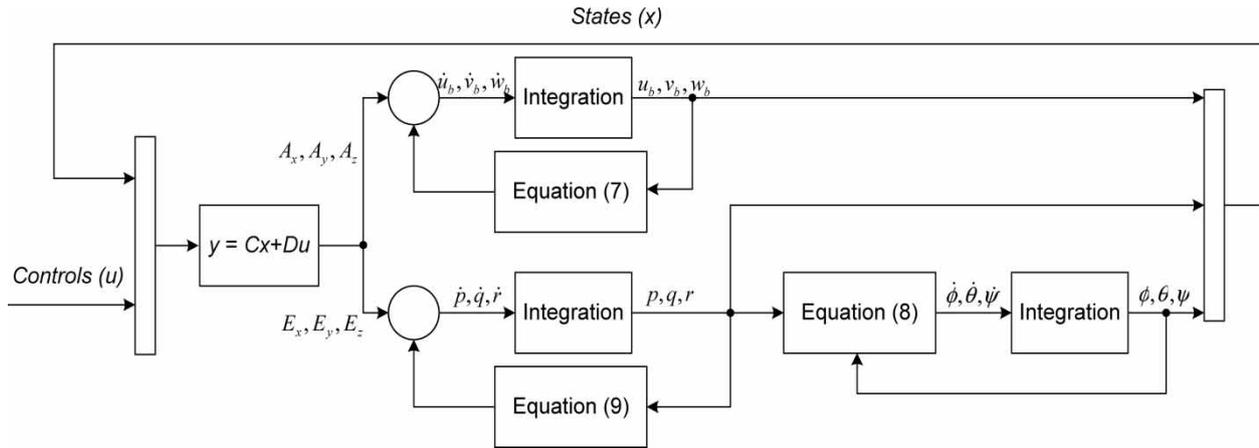


Fig. 6 The linear aircraft model and the six-DOF rigid body

$N$  is the number of recorded samples,  $N_0$  is the number of outputs = 6, and  $N_R$  is the number of records = 4, one for each applied control.

During the optimization procedure the four records corresponding to each control excitation are separately considered, and not concatenated together

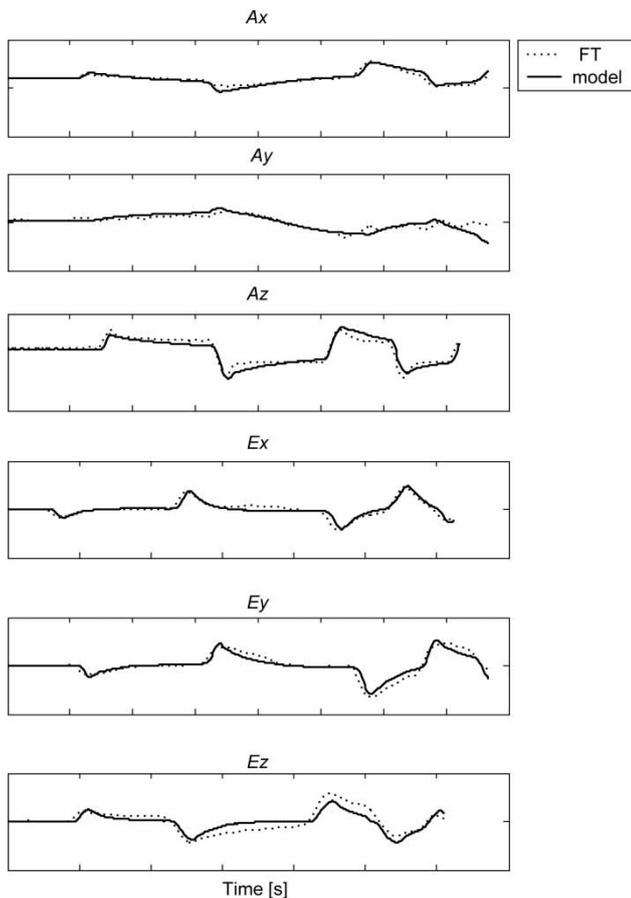


Fig. 7 Comparison between the flight test and the generated model

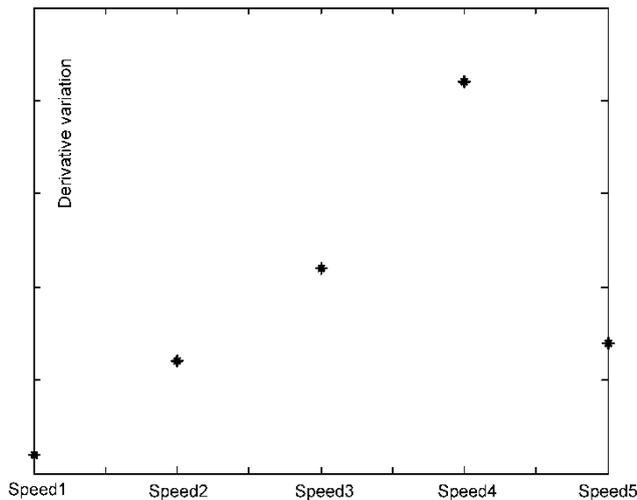
as in Fig. 1. In this way, the trim of each record is automatically taken into consideration during the optimization process. The jumps artificially created by concatenation are eliminated, thus, not modelled.

The system response following the optimization is presented in Fig. 7. In the legend of Fig. 7, FT represents the flight test data and model represents our model output. The differences between results shown in Figs 5 and 7 (the system response before and after optimization) are due to the fact that the system does not properly perform before the optimization. The results in Fig. 5 are shown for a much shorter period of time than results shown in Fig. 7, because of their divergence on a longer period of time (without optimization algorithm). However, after optimization, as shown in Fig. 7, the system is simulated during the whole length of time record. After optimization, new matrices  $C$  and  $D$  are obtained, representing the model for the considered flight conditions and the seed for future estimations, for other flight conditions.

## 6 MODEL VALIDATION

The case presented in the above paragraphs represents one flight condition, defined by the following five parameters: indicated airspeed, the gross weight, the position of the centre of gravity, the altitude, and the vertical velocity. Each element of  $C$  and  $D$  matrices represents, in fact, a point in a five-dimensional space defined by these parameters.

The elements of  $C$  matrix, which multiply the state parameters  $x$  in equation (3), are called stability derivatives, while the elements of  $D$  matrix, which multiply the control inputs  $u$ , are called control derivatives. The derivatives vary as function of flight conditions, defined by the five parameters. An example of variation of one of the derivatives as function of indicated airspeed is presented in Fig. 8. The values of



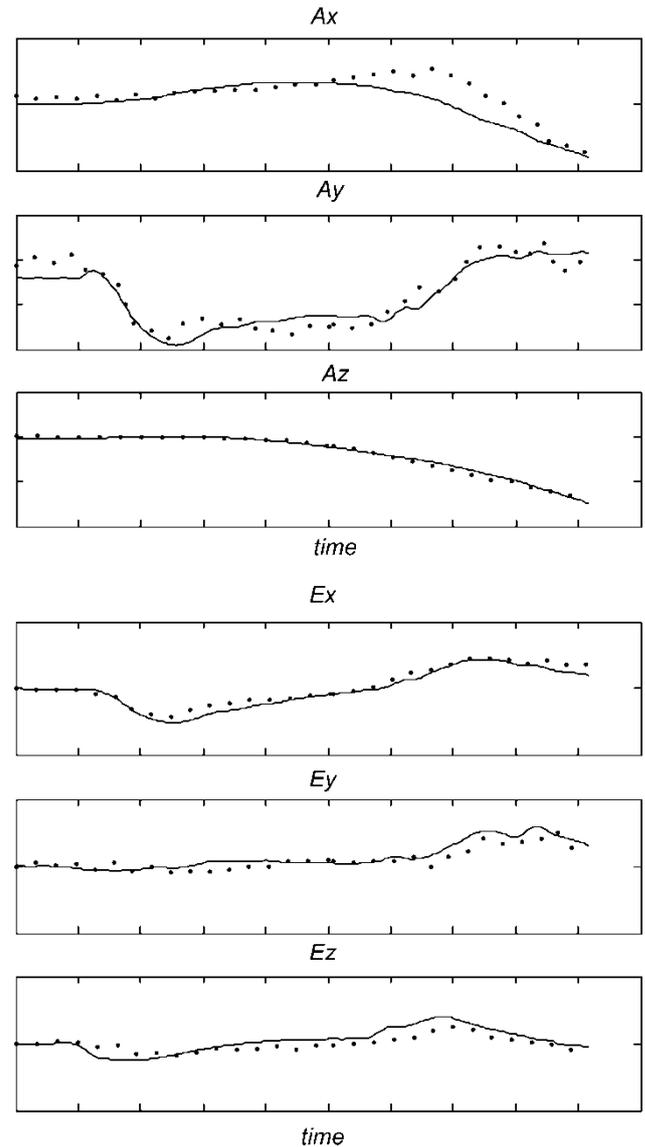
**Fig. 8** Variation of a derivative as function of indicated airspeed

indicated airspeed are from 30 knots up to 90 per cent of the helicopter maximum speed. The actual values of derivatives presented in Fig. 8 are confidential and not essential for the algorithm developed in the paper. Further development of the aircraft model is finding the approximation functions describing the dependencies of each element of **C** and **D** matrices as functions of the five parameters.

The next step would be the model validation using qualifying tests. These tests consist in a collection of a set of flight test data that will be used to qualify the flight dynamics model in accordance with FAA AC 120-63. These data include:

- performance in level flight and in hover;
- controllability;
- general flight characteristics;
- general stability.

For a complete validation the whole model is needed, for all flight conditions, and the five-dimensional approximation function. However, the development of whole model is not the topic of the present paper, which just presents a new identification technique that eliminates some of the drawbacks met during the model development by use of MLE. Even without developing the whole model, its validity can still be checked using one of the qualifying tests. The one that is randomly chosen is a longitudinal manoeuvring stability check using a lateral control inducing a lateral acceleration of 1.2 g. The qualifying test was performed for the same gross weight, CG position, altitude, and vertical velocity. The indicated airspeed is the only parameter that was varied. The value the qualifying test was performed is in the middle of Speed2-Speed3 interval. For this reason, it was called Speed23. Due to the fact that the approximation functions for stability and control derivatives were



**Fig. 9** Example of model qualification test

not computed, the model validity for both models, developed at Speed2 and the Speed3, was checked. Although the two simulations did not give the same results, the differences were not large enough to justify the presentation here of both of them. Therefore, Fig. 9 presents only the results of the simulation using the flight test data at Speed23 and the response of the model created for Speed2. The small error obtained by employing two different models developed for different flight conditions shows the robustness of the model built using the proposed methodology.

## 7 CONCLUSIONS

In the current paper, a new algorithm is presented to compute an aircraft model, in time-domain. The identification procedure uses a fuzzy approach to generate

the starting matrices used in optimization. The convergence is fast and the generated model is robust, fitting other flight conditions which, in previously developed model, needed a separate model. Other flight conditions are estimated without using other information than the flight test data.

The advantages of the proposed algorithm with respect to classical estimation parameters theories such as MLE are the following.

1. There is no need of initial matrices calculations; MLE needs seed matrices that are meaningful, from another similar helicopter or from the same helicopter for different flight tests conditions, and does not converge for far fetched matrices.
2. The convergence of estimation procedure is faster; The MLE convergence process is slow, and for this reason, the MLE does not converge for far fetched matrices. The optimization speed is dependent of the initial conditions (initial matrices). The algorithm here presented needs less than 4 min to compute a linear model without any initial matrices and less than 3 min to compute a model by re-using matrices from a previously computed flight condition. It takes less than 2 days for computing the models for the whole flight envelope, which here consisted in more than 200 points. For comparison, in MLE case 2 days might be not enough for finding an appropriate seed matrix; therefore, the whole process of punctual models generation took more than 2 months.
3. The state space variables are fully controlled; MLE has also control on state space variables. Other methods, such as Matlab system identification toolbox methods, do not give good results when trying to control the state variables. Good results, in term of curve fitting, are obtained through when the algorithm itself chooses the state parameters, in which case, they are not the linear and angular speeds, but physical meaningless states.
4. The non-linear dependencies between parameters can be added; although the subject of future developments, the second-order non-linearities of the states products types can be easily added to the linear model and the coefficients of these products can be optimized following the same procedure as the one shown in section 5.
5. The generated model is more robust; tested in different flight conditions, the computed matrices are very close; the model covers a high number of flight conditions, reducing in this way the number of flight tests needed for its development which gives flight test cost reductions. The algorithm can be applied to different flight tests, for instance to the qualification tests in Fig. 9. The results show the values of changes in derivatives values that would fit the qualification tests. The derivatives values

obtained from qualification tests could be used as supplementary points on Fig. 8.

The entire algorithm can be implemented using Matlab<sup>®</sup>, and the simulations are performed in Simulink<sup>®</sup> environment.

While developing flight models for an aircraft using MLE showed its limitations establishing the correct trim conditions during proof-of-match (POM) performance. Indeed, putting together four records, one after another, corresponding to four consecutive excitations of aircraft controls leads to some jumps in the time histories, caused by the fact that the end of one record does not match the trim of the next one. The disadvantage of MLE is that maximizing the resemblance of the model's output to the measured data it includes these fake jumps into the model. Therefore, a modelling of these artificial jumps is performed, and they do not reflect a reality of the aircraft evolutions. The trim of the MLE model is affected, because about 2 months was spent to set up the trim functions for the model estimated using this algorithm. As one can see in the Fig. 9, in our approach, the optimization is performed with respect to each record trimmed separately. Therefore, no difficulty was encountered in trimming the state-space model.

Other drawbacks of the MLE approach are related to the use of the first equation (3) to compute the states, using **A** and **B** matrices, therefore an approximate linear model. In these conditions, MLE gives very good match when the data used to generate the model are fed to the input, and when the states are computed using **A** and **B** matrices. However, during POM the non-linear equation (7), is used to compute the states. The match is not that good anymore and analysis of differences by an experimented operator is mandatory. Small changes of the elements of **C** and **D** matrices are performed, in order to improve the match. This operation is automatically performed during the optimization phase, obtaining a substantial reduction of the time needed to achieve the match in within Federal Aviation Administration tolerances for level D simulators. The MLE algorithm is optimal to retrieve **A**, **B**, **C**, **D** matrices that maximize the likelihood of the modelled outputs to the desired ones. However, in the steps following finding the stability and control derivatives, namely finding the approximation function and changing the model during POM process, one breaks down the resemblance, and consequently the optimality. In this approach the optimality is given from the beginning, looking for a model good enough to be used in the following steps of linearization and optimization. By doing so, a model that needs little adjustments during POM was created from beginning, greatly reducing the necessary time for this process.

As future development of this strategy to estimate a model, the possibility of adding non-linear dependencies is mentioned. If the match presented in Fig. 9 is considered as the best obtained with a linear model, then the differences between measured and estimated parameters are caused by non-linearities. For instance, the second-order non-linearities are  $x^2$ ,  $u^2$ , and  $ux$ . Check of the correlations between these terms and the error for each record in Fig. 7 would allow to select which one of these non-linearities has the highest influence and add it into the model. The weight coefficients are to be determined using an optimization technique. In the end, this new method is recommended to be used for aircraft and helicopter flight tests validation.

## REFERENCES

- 1 Paris, A. C. and Bonner, M. Nonlinear model development from flight-test data for F/A – 18E super hornet. *AIAA J. Aircr.*, 2004, **41**(4), 692–702.
- 2 Theodore, C. R., Tischler, M. B., and Colbourne, J. D. Rapid frequency-domain modeling methods for unmanned aerial vehicle flight control applications. *AIAA J. Aircr.*, 2004, **41**(4), 735–743.
- 3 Shroeder, M. R. Synthesis of low-peak-factor signals and binary sequences with low autocorrelation. *IEEE Trans. Inf. Theory*, 1970, **16**(1), 85–89.
- 4 NATO – AGARD. Identification of dynamic systems – application to aircraft. Part 2: nonlinear analysis and manoeuvre design, 1994.
- 5 Sugeno, M. *Industrial applications of fuzzy control*, 1985 (Elsevier Publication, Amsterdam).
- 6 Hagan, M. T., Demuth, H. B., and Beale, M. H. *Neural network design*, 1996 (PWS Publishing Company, Boston, MA).
- 7 Nelder, J. A. and Mead R. A simplex method for function minimization. *Comput. J.*, 1965, **7**, 308–313.

## APPENDIX

### Notation

<b>A, B, C, D</b>	state-space matrices	$A_z$	linear acceleration in the $z$ direction
$A_x$	linear acceleration in the $x$ direction	$E_x$	angular acceleration in the $x$ direction
$A_y$	linear acceleration in the $y$ direction	$E_y$	angular acceleration in the $y$ direction
		$E_z$	angular acceleration in the $z$ direction
		$i, j$	summation indexes
		$I$	helicopter inertia tensor
		$J$	the objective function to be minimized
		meas	index standing for measured parameters
		model	index standing for simulated parameters
		$N$	the number of recorded samples
		$N_O$	number of outputs
		$N_R$	number of records
		$p$	angular rate around $x$ -body axis
		$P$	angular rate around $x$ -body axis at trim
		$q$	angular rate around $y$ -body axis
		$Q$	angular rate around $y$ -body axis at trim
		$r$	angular rate around $z$ -body axis
		$R$	angular rate around $z$ -body axis at trim
		$u$	input vector containing the aircraft controls
		$u_b$	linear velocity along $x$ -body axis
		$U_b$	linear velocity along $x$ -body axis at trim
		$v_b$	linear velocity along $y$ -body axis
		$V_b$	linear velocity along $y$ -body axis at trim
		$w_b$	linear velocity along $z$ -body axis
		$W_b$	linear velocity along $z$ -body axis at trim
		$x$	nine components state vector containing linear velocities, angular velocities, and Euler angles
		$y$	six components output vector containing linear and angular accelerations
		$\alpha$	angle-of-attack
		$\beta$	sideslip angle
		$\phi, \theta, \psi$	Euler angles

