Aircraft longitudinal stability and control derivatives identification by using life cycle and Levenberg–Marquardt optimization algorithms

Felipe Antonio Chegury Viana, Benedito Carlos de Oliveira Maciel, Nei Salis Brasil Neto, Marcelo Fernandes de Oliveira, Valder Steffen Jr & Luiz Carlos Sandoval Góes

To cite this article: Felipe Antonio Chegury Viana, Benedito Carlos de Oliveira Maciel, Nei Salis Brasil Neto, Marcelo Fernandes de Oliveira, Valder Steffen Jr & Luiz Carlos Sandoval Góes (2009) Aircraft longitudinal stability and control derivatives identification by using life cycle and Levenberg–Marquardt optimization algorithms, Inverse Problems in Science and Engineering, 17:1, 17-34, DOI: 10.1080/17415970802082773

To link to this article: https://doi.org/10.1080/17415970802082773

Published online: 27 Nov 2008.

Article views: 57

View related articles

Citing articles: 2 View citing articles
Aircraft longitudinal stability and control derivatives identification by using life cycle and Levenberg–Marquardt optimization algorithms

Felipe Antonio Chegury Viana\textsuperscript{a}, Benedito Carlos de Oliveira Maciel\textsuperscript{b}, Nei Salis Brasil Neto\textsuperscript{b}, Marcelo Fernandes de Oliveira\textsuperscript{b}, Valder Steffen Jr.\textsuperscript{a} and Luiz Carlos Sandoval Gôes\textsuperscript{b}\textsuperscript{*}

\textsuperscript{a}School of Mechanical Engineering, Federal University of Uberlândia, Uberlândia, Brazil; 
\textsuperscript{b}Department of Mechanical-Aeronautical Engineering, Technological Institute of Aeronautics, São José dos Campos, Brazil

(Received 16 April 2007; final version received 30 November 2007)

In this work, two optimization algorithms are investigated to accomplish the parameter identification of the longitudinal motion of a real aircraft by using the output error method. The first algorithm is the nature-inspired algorithm named the life cycle model, which is a composed strategy based on other heuristics such as genetic algorithms and particle swarm optimization. The second one is the gradient-based technique named Levenberg–Marquardt algorithm, which is a variant of the Gauss–Newton method. Flight test data, performed with a training jet aircraft (Xavante AT-26), were used to feed the output error method. In this context, both optimization algorithms were tested, in solo performance and in a cascade-type approach. Results are reported, aiming to illustrate the success of using the proposed methodology.

\textbf{Keywords:} system identification; optimization; parameter estimation; evolutionary algorithms

1. Introduction

System identification is a scientific discipline that provides answers to the inverse problem of obtaining a description in some suitable form for a system, given its behaviour as a set of observations [1]. Considering that one of the most important phases of the modern design and evaluation process concerns modeling and simulation, the system identification and parameter estimation constitutes a fundamental step to achieve this goal. System identification is a general procedure used to match the observed input–output response of a dynamic system by a proper choice of an input–output model and respective physical parameters. From this point of view, the aircraft system identification or inverse modelling comprises of the proper choice of aerodynamic models, the development of parameter estimation techniques by optimization of the mismatch error between predicted and real aircraft response and the development of proper tools for integration of the equations of motion within the system simulation and correlated activities [2].

*Corresponding author. Email: goes@ita.br
The importance of obtaining flight-derived aircraft parameters was recognized early in aircraft evolution. A couple of years after the introduction of the classical stability approach by Bryan [3], Glauert’s work in 1919 on the analysis of phugoid motion and that of Norton during 1919–1923 on estimating a number of derivatives mark the beginning of the experimental investigation of dynamic stability from actual flight data [4,5]. During the late 1940s and early 1950s, several techniques were introduced, such as steady-state oscillatory excitations by Milliken [6], pulse-input methods incorporating Fourier analysis by Seamans [7], weighted least-squares, and response curve fitting by Shinbrot [8]. An excellent account of dynamic stability and control during this early period is found in [9] and a survey of methods for determining stability and control derivatives from dynamic flight measurements in [10].

The techniques of the 1940s to early 1960s were mostly frequency response methods, and were either limited to the estimation of incomplete sets of coefficients, to simple motions, or restricted for other reasons.

Although Shinbrot had introduced the response curve fitting method, which is equivalent to today’s output error method, it was at that time found to be impracticable owing to the lack of adequate (digital) computational means. During the early 1960s, analogue matching was a popular technique of updating and validating wind-tunnel predictions of stability derivatives based on flight tests.

The modern era of system identification is marked by the implementation of the maximum-likelihood method on a digital computer by Astrom and Bohlin in 1965, and its application to an industrial plant represented by difference equations [11]. The automatic data processing capability provided by digital computers dramatically changed the focus of flight data analysis from frequency-domain methods to time-domain methods. It became possible to obtain a significantly larger number of stability and control derivatives from a single flight test.

Since then, the flight vehicle system identification gradually evolved to a coordinated approach based on flight test techniques, flight test instrumentation and methods of data analysis. All these strongly interdependent topics can be summarized in four important aspects of the art and science of system identification, namely manoeuver, measurements, methods and models. These ‘Quad-M’ requirements must be carefully investigated for each flight vehicle, and are the key to successful flight vehicle system identification [1].

The present work is mainly concerned with the methods aspects and reports the identification of longitudinal stability derivatives from flight test data by using one method in particular, which is the output error method [12,13].

It is well known that the identification by using output error method associated with classical gradient-based optimization methods is a difficult task due to the existence of local minima in the design space. Moreover, such methods require an initial guess to the solution and it is not possible to assure global convergence. These aspects have motivated the authors of this article to explore the performance of a gradient-based, namely the Levenberg–Marquardt method (LM) [13], and a heuristic method, namely the life cycle model (LC) [14]. LM is a second-order variant of the Gauss–Newton method and LC is a hybrid approach, based on two heuristic methods, namely genetic algorithms (GA) and particle swarm optimization (PSO) combined in an evolutionary strategy. The individual performance of both algorithms is tested and then a cascade-type scheme is proposed.
aiming to take advantage of the global search capabilities of LC and the local search capabilities of LM.

An optimization problem is formulated so that the objective function represents the difference between the measured characteristics of the structure and its model counterpart. From a linear state space model for the longitudinal motion of aircraft, a total number of 10 parameters are assumed as being unknown. In this case, the parameters represent the dimensional aerodynamic coefficients of the aircraft.

In order to show the optimization strategy and the approaches, experimental flight test data, performed with a military aircraft (Xavante AT-26), were used in parameter identification of the stability derivatives.

2. Problem formulation

2.1. Aircraft longitudinal model

In this work the inverse problem formulation is applied to the longitudinal equations of movement of the aircraft, in particular, the short period mode, for which the linear state space equations can be written as:

\[
\begin{bmatrix}
\dot{\alpha} \\
\dot{q}
\end{bmatrix} = \begin{bmatrix}
Z_\alpha & (1 + Z_\alpha) \\
M_\alpha & M_q
\end{bmatrix} \begin{bmatrix}
\alpha \\
q
\end{bmatrix} + \begin{bmatrix}
Z_{\delta e} \\
M_{\delta e}
\end{bmatrix} \delta e + \begin{bmatrix}
b_{x_1} \\
b_{x_2}
\end{bmatrix},
\]

and the observation equations are given by:

\[
\begin{bmatrix}
y_1 \\
y_2
\end{bmatrix} = \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix} \begin{bmatrix}
\alpha \\
q
\end{bmatrix} + \begin{bmatrix}
0 \\
0
\end{bmatrix} \delta e + \begin{bmatrix}
b_{y_1} \\
b_{y_2}
\end{bmatrix}.
\]

The previous dynamical equation has 10 unknown parameters that need to be estimated, giving \( \Theta \in \mathbb{R}^{10} \):

\[
\Theta = [Z_\alpha, Z_q, M_\alpha, M_q, Z_{\delta e}, M_{\delta e}, b_{x_1}, b_{x_2}, b_{y_1}, b_{y_2}]^T.
\]

These parameters \( Z_\alpha, Z_q, M_\alpha, M_q, Z_{\delta e}, M_{\delta e} \) are the stability and control derivatives, which represent the aerodynamic model of the aircraft. The constant parameters \( b_{x_1}, b_{x_2}, b_{y_1} \) and \( b_{y_2} \) comprise measurement biases in the output and input variables, and initial conditions on the state variables.

2.2. Output error method

In this section, the parametric identification, in particular the parameter estimation applied to a linear model of the longitudinal motion of an aircraft in space state formulation is presented. The output error method, as shown in Figure 1, is one of the most used estimation methods for aerodynamic parameter estimation from flight test data [15–17]. It has several desirable statistical properties, including its applicability to non-linear dynamical systems and the proper accounting of measurements noise [16].

The structure of the model is considered to be known and, in this case, is the one shown in Equations (1) and (2). So, the identification process consists of determining the
parameter vector $\Theta$, which gives the best prediction of the output signal $y(t)$, using an optimization criteria. The attainment of an estimate through optimization of a cost function based on the prediction error of the plant requires, usually, the minimization of a non-linear function. Thus, LC, LM or a composition of both is used here to estimate the parameters in the model. Therefore, the cost function to be minimized involves the prediction error:

$$e(k) = \hat{y}(k) - y(k), \quad (4)$$

where $\hat{y}(k)$ is the output prediction based on the actual estimate $\hat{\Theta}$ of the parameter vector $\Theta$.

Consider a dynamic system, identifiable, with a defined model $M(\Theta)$ and output $y$. Suppose that $p(y | \Theta)$ is the conditional probability Gaussian distribution of the random variable $y$ with dimension $m$, mean $f(\Theta)$, and covariance $R$, with dimension $m \times m$. $p(y | \Theta)$ is known as the likelihood functional. Goodwin and Payne in [12] attribute this name due to the fact that it is a measure of the probability of occurrence of the observation $y$ for a given parameter $\Theta$. Thus, the likelihood functional is:

$$p(y | \Theta) = \frac{1}{(2\pi)^{m/2} |R|^{n/2}} \exp \left\{ -\frac{1}{2} \sum_{k=1}^{n} [e(k, \Theta)]^T [R^T]^{-1} [e(k, \Theta)] \right\}. \quad (5)$$

The maximum likelihood estimate (MLE) is defined as the value of $\Theta$ which maximizes this functional, in such a way that the best estimate of $\Theta$ is:

$$\hat{\Theta} = \text{ArgMax} \, p(y | \Theta). \quad (6)$$

The maximization of $p(y | \Theta)$ is equivalent to the minimization of the functional $J(\Theta)$, which is given by:

$$J(\Theta) = \sum_{k=1}^{n} \frac{1}{2} \left[ [e(k, \Theta)]^T [R^T]^{-1} [e(k, \Theta)] + \ln |R| \right]. \quad (7)$$

Thus, in the optimization process, $J(\Theta)$ is equivalent to $-\ln(p(y | \Theta))$, except for a constant term.

---

Figure 1. Block diagram of the estimation procedure.
2.3. Optimization strategies
As can be seen in the general framework illustrated by Figure 1, the output error has no specific requirement about the optimization method. Even though most works in the literature report the use of classical methods, there are no limitations in using other approaches. In this scenario, three optimization strategies, described in Table 1, are explored to solve the identification problem.

3. Life cycle model

3.1. Back to nature
Life cycle model (LC) is a computational tool inspired by the biologic concept of the life cycle. From a biology viewpoint, the term is used here to define the passage through the phases during the life of an individual. Some phases, such as sexual maturity, are one-time events, others, such as mating seasons, are re-occurring. Although it does not happen in every case, the transitions between life cycle phases are started by environmental factors or by the necessity to fit to a new condition [14]. The transition process promotes the maturity of an individual and contributes to the adaptation and evolution of its species.

Take the example of a honeybee swarm. Even though 10,000 individuals exist in a colony, there are just three castes of bees:

- queens, which produce eggs,
- drones or males, which mate with the queen, and
- workers, which are all non-reproducing females.

Table 2 shows the life cycles of each bee type in a honeybee colony.

After this simple example, it is easy to visualize the concept of life cycle.

3.2. Algorithm description
From an implementation point of view, algorithms such as GA and PSO are heuristics search methods of proven efficiency as optimization tools. LC intends to put together the positive characteristics found in each method and creates a self-adaptive optimization approach. Each individual, as a candidate solution, decides based on its success if it would prefer to belong to a population of a GA, or to a swarm of a PSO. This means that

Table 1. Optimization strategies used in the output error scheme.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Optimization method</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Life cycle model</td>
<td>The main explored characteristics are the global search capability and the low level of knowledge about the search space.</td>
</tr>
<tr>
<td>2</td>
<td>Levenberg–Marquart</td>
<td>As a classical method, local search capabilities and small number of function evaluations are the main explored advantages.</td>
</tr>
<tr>
<td>3</td>
<td>Life cycle model and Levenberg–Marquart</td>
<td>The result of a first global search performed by LC is used as an initial guess in the LM method. This cascade approach aims to make use of the main advantages of both methods.</td>
</tr>
</tbody>
</table>
different heuristic techniques contribute together to form a robust high-performance optimization tool. The idea is that complex problems can be conveniently considered from the optimization viewpoint. As can be seen, the less successful individuals must change their status in order to improve their fitness. In plain English, the optimization approach does not follow a rigid scheme as proposed in [19], in which various techniques are used sequentially in a cascade-type structure. In other words, it is the mechanism of self-adaptation of the optimization problem that rules the procedure. Figure 2 shows the outline of a basic LC algorithm.

Take the example of an LC with just GA and PSO as heuristics. According to Figure 2, the algorithm is initialized with a set of particles of a PSO swarm, which can turn into GA individuals, and then, according to their performance, return to particles again, and so on. An LC individual switches its stage when there is no fitness improvement for more than a previously defined number of iterations. This is a parameter that can be adjusted according to the problem.

3.3. Algorithm parameters
Since the algorithm is composed by various heuristics, it is necessary to set the parameters of every heuristic used in the life cycle. Nevertheless, there is a parameter inherent to the life cycle, namely the number of iterations that represents a stage of the life cycle. The authors, to improve the algorithm performance, have introduced this parameter differently from what is usually done, by simply fixing this number as in [14]. This number is called stage interval. At the end of each stage interval, the less successful individuals must change their stage aiming at improving their fitness.

More detailed information about LC can be found in [14,20].

4. Genetic algorithms
A genetic algorithm is an optimization algorithm based on Darwin’s theory of survival and evolution of species. The algorithm starts from a population of random individuals, viewed as candidate solutions to the problem. During the evolutionary process, each individual in the population is evaluated, reflecting its adaptation capability to the environment. Some of the individuals in the population are preserved while others are discarded. This process mimics the natural selection in Darwinism. The remaining group
of individuals is paired in order to generate new individuals to replace the worst ones in the population, which are discarded in the selection process. Finally, some of them can be subjected to mutation, and as a consequence, the chromosomes of these individuals are altered. The entire process is repeated until a satisfactory solution is found. The outline of a basic GA is as follows (see Figure 3).

Although the initial proposed GA was dedicated to discrete variables, nowadays, improvements are available to deal with discrete and continuous variables, see [21,22] for more details.

5. Particle swarm optimization
Particle swarm optimization was introduced by Kennedy and Eberhart in [23]. The method reflects the social behaviour of some bird species. The main idea of PSO is to reproduce the search procedure used by a swarm of birds when looking for food
or a nest. This is achieved by modelling the flying using a velocity vector, somewhat analogous to a search direction. The velocity vector considers a contribution of the current velocity and other two portions referred to the knowledge of the particle itself and of the swarm, respectively, about the search space. In such a way, the velocity vector is used to update the position of each particle in the swarm. The outline of a basic PSO is as follows (see Figure 4):

PSO is comprehensively presented in [23,24].

6. Levenberg–Marquardt optimization method
The Levenberg–Marquardt method is a second-order technique based on the Gauss–Newton method. This method, although complex, is suitable for a quadratic cost function, and is expected to converge quickly. First, we approximate $J(\Theta)$ by a parabolic function $J_L(\Theta)$ under the condition $\Theta_L$ (retaining only the three first Taylor series terms):

$$J_L(\Theta) \approx J_L(\Theta_L) + (\Theta - \Theta_L)^T \nabla^2 J(\Theta_L) + \frac{1}{2} (\Theta - \Theta_L)^T \nabla^2 J(\Theta_L) [\nabla^2 J(\Theta_L)] (\Theta - \Theta_L).$$

Figure 3. Basic scheme for GA optimization.
The optimization condition is obtained when:

$$\nabla_{\Theta} J(\Theta^*) = 0.$$  (9)

Applying Equation (9) to Equation (8):

$$\nabla_{\Theta} J_L(\Theta) = \nabla_{\Theta} J(\Theta_L) + (\Theta - \Theta_L)^T \left[ \nabla^2_{\Theta} J(\Theta_L) \right] = 0,$$  (10)

which can be used to find the optimum, $\Theta^*$, through the recursion:

$$\Theta_{i+1} = \Theta_i - \left[ \nabla^2_{\Theta} J(\Theta_i) \right]^{-1} \nabla_{\Theta} J(\Theta_i).$$  (11)

The complexity in the calculation of the Hessian matrix, $\nabla^2_{\Theta} J(\Theta)$, in Equation (11), is avoided through the Gauss–Newton method, which uses the approximation:

$$\nabla^2_{\Theta} J(\Theta) \approx \sum_{k=1}^{n} \left[ \nabla_{\Theta} \hat{y}(\Theta) \right]^T \left[ \hat{R} \right]^{-1} \left[ \nabla_{\Theta} \hat{y}(\Theta) \right],$$  (12)

where the terms involving the second derivative are discarded and the gradient of the estimated output, $\nabla_{\Theta} \hat{y}(\Theta)$, is called sensibility function.

LM is an extension of the Gauss–Newton algorithm [13]. The idea is to modify Equation (12) to:

$$\nabla^2_{\Theta} J(\Theta) \approx \sum_{k=1}^{n} \left[ \nabla_{\Theta} \hat{y}(\Theta) \right]^T \left[ \hat{R} \right]^{-1} \left[ \nabla_{\Theta} \hat{y}(\Theta) \right] + \lambda I,$$  (13)

where the inversion of the matrix is not performed explicitly, but solving by using singular value decomposition (SVD) we get the following expression:

$$[\nabla^2_{\Theta} J(\Theta) + \lambda I] \Delta \hat{\Theta} = \nabla_{\Theta} J(\Theta).$$  (14)
The addition of the term $\lambda I$ in Equation (14) solves the problem of ill-conditioning of the matrix. The Levenberg–Marquardt algorithm can be interpreted in the following manner: for small values of $\lambda$, it behaves like the Gauss–Newton algorithm, while for high values of $\lambda$, it behaves like the steepest gradient algorithm.

7. Experimental tests

The identification of longitudinal stability derivatives from flight test data was performed by using a set of experimental flight data from a military aircraft. The aircraft used was the Xavante – AT-26 located at the Special Flight Test Group of the CTA (GEEV-CTA), as illustrated in Figure 5. Table 3 reports some geometric and inertial parameters of the aircraft.

The AT-26 was equipped with an instrumentation system in order to collect flight test data for the system identification experiments of this work. Generally, the measurements provided by the flight test instrumentation are introduced to the identification algorithm through the observation equations, as stated in Equation (2).

So, for the present work, the AT-26 flight test instrumentation system provides measurements of the following variables: angle-of-attack, pitch rate and elevator deflection.

In a general manner, the instrumentation system is composed of flight sensors, data acquisition and record unity. The flight sensors that measure the aforementioned variables (angle-of-attack, pitch rate and elevator deflection) have different natures.

![Figure 5. Xavante AT-26 training aircraft.](image)

Table 3. Xavante’s geometric and inertial parameters.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>$m_0$</th>
<th>$I_y$</th>
<th>$c$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>4210</td>
<td>6500</td>
<td>1.90</td>
<td>20.62</td>
</tr>
<tr>
<td>Unit</td>
<td>kg</td>
<td>kg m$^2$</td>
<td>m</td>
<td>m$^2$</td>
</tr>
</tbody>
</table>
The angle-of-attack, being an airflow measurement, requires an air data probe installed in a region of the aircraft in which the minimum disturbance of the inflow air by the aircraft is encountered. This variable is generally measured by vanes and potentiometers. The pitch rate, on the other hand, being an inertial quantity, requires an inertial measurement unit generally installed near the centre-of-gravity of the air vehicle in order to reduce the necessity of some corrections in acceleration measurements. The inertial measurement unit used in the AT-26 flight test instrumentation system is equipped with micro-electro-mechanical gyrometers and accelerometers. The elevator deflection, being a displacement, is generally measured by potentiometers directly installed in the aircraft control surfaces. Table 4 gives some information about the AT-26 measurements of interest for this work.

For all measurements the existence of systematic errors is assumed, the so-called bias error. As mentioned in section ‘aircraft longitudinal model’, these parameters can also be identified. They take into account some installation and/or sensors inaccuracies during the data acquisition procedure.

A unique longitudinal flight test manoeuvre for collecting the data was used in the present work for identification purposes. The flight test manoeuvre was manually applied by the pilot. It was specified in order to excite, mainly, the short period longitudinal dynamic model. Figure 6 shows the elevator deflection measured by the elevator potentiometer during this manoeuvre.

8. Results

Table 5 gives the setup for LC used in the identification tests. It is important to point out that the aerodynamic derivatives are a parameter to be estimated as well as the sensors bias errors.

Given the random nature of LC and the dependency of the initial guess of LM, the results obtained using strategy 1 or 2 may vary from one run to another. This way, for the sake a reasonable statistical analysis, each one of the three strategies was repeated 20 times and the best, worst, average, standard deviation, and coefficient of variation of the results from each of the 20 repetitions were recorded. For strategies 1 and 2, LC and LM were set with different and random initial populations and initial guesses, respectively. For strategy 3, the results obtained when running strategy 1 were used as initial guesses for LM. The average indicates the central tendency which the heuristic results tend to. The standard deviation indicates the stability of the

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Acquisition rate (hz)</th>
<th>Acquisition accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle-of-attack</td>
<td>32</td>
<td>0.11</td>
</tr>
<tr>
<td>Pitch rate</td>
<td>32</td>
<td>0.5</td>
</tr>
<tr>
<td>Elevator deflection</td>
<td>16</td>
<td>1.0</td>
</tr>
</tbody>
</table>
optimal solutions, measuring the degree of convergence. In order to express this convergence degree regardless of the magnitude of each of the responses, the coefficient of variation is calculated through the division of the standard variation by the mean value of the optima. It is worth mentioning that the average value for $J$ is the average of values of the functional over the 20 runs; it does not correspond to the value obtained when running the analysis with the average design. Tables 6–8 report a set of identification results obtained following the three different optimization strategies, as presented in Table 1.

Figure 7 helps the statistical interpretation of the results. Comparing all metrics of the functional $J$, it is easy to see the advantages of strategy 3. For strategy 1, Table 6 and see

---

**Table 5. Life cycle parameters used in the experimental identification.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Search space</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower bound</td>
</tr>
<tr>
<td>$Z_a$</td>
<td>$-5$</td>
</tr>
<tr>
<td>$Z_q$</td>
<td>0</td>
</tr>
<tr>
<td>$M_d$</td>
<td>$-5$</td>
</tr>
<tr>
<td>$M_q$</td>
<td>$-5$</td>
</tr>
<tr>
<td>$Z_{sc}$</td>
<td>0</td>
</tr>
<tr>
<td>$M_{sc}$</td>
<td>$-10$</td>
</tr>
<tr>
<td>$b_{s1}$</td>
<td>$-0.1$</td>
</tr>
<tr>
<td>$b_{s2}$</td>
<td>$-0.1$</td>
</tr>
<tr>
<td>$b_{y1}$</td>
<td>$-0.1$</td>
</tr>
<tr>
<td>$b_{y2}$</td>
<td>$-0.1$</td>
</tr>
<tr>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>200</td>
</tr>
<tr>
<td>Stage interval</td>
<td>20</td>
</tr>
</tbody>
</table>
Table 6. Identification results for LC (strategy 1).

<table>
<thead>
<tr>
<th></th>
<th>Best</th>
<th>Worst</th>
<th>Average</th>
<th>Standard deviation</th>
<th>Coefficient of variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_a$</td>
<td>$1.9439$</td>
<td>$1.3795$</td>
<td>$2.2321$</td>
<td>$0.9011$</td>
<td>$-0.4037$</td>
</tr>
<tr>
<td>$Z_q$</td>
<td>$1.0040$</td>
<td>$1.1927$</td>
<td>$0.9939$</td>
<td>$0.4068$</td>
<td>$0.4093$</td>
</tr>
<tr>
<td>$M_a$</td>
<td>$-2.8499$</td>
<td>$0.0076$</td>
<td>$-1.4184$</td>
<td>$0.7578$</td>
<td>$-0.5343$</td>
</tr>
<tr>
<td>$M_q$</td>
<td>$-0.8256$</td>
<td>$2.9405$</td>
<td>$2.5743$</td>
<td>$0.9441$</td>
<td>$0.3667$</td>
</tr>
<tr>
<td>$b_{x1}$</td>
<td>$0.0988$</td>
<td>$0.0762$</td>
<td>$0.0950$</td>
<td>$0.0079$</td>
<td>$0.0834$</td>
</tr>
<tr>
<td>$b_{y1}$</td>
<td>$0.0964$</td>
<td>$0.0659$</td>
<td>$0.0868$</td>
<td>$0.0113$</td>
<td>$0.1298$</td>
</tr>
<tr>
<td>$b_{x2}$</td>
<td>$0.0381$</td>
<td>$0.0945$</td>
<td>$0.0621$</td>
<td>$0.0189$</td>
<td>$0.3053$</td>
</tr>
<tr>
<td>$J$</td>
<td>$7.4842 \times 10^{-9}$</td>
<td>$4.3593 \times 10^{-8}$</td>
<td>$2.1368 \times 10^{-8}$</td>
<td>$8.7851 \times 10^{-9}$</td>
<td>$0.4111$</td>
</tr>
</tbody>
</table>

Table 7. Identification results for LM (strategy 2).

<table>
<thead>
<tr>
<th></th>
<th>Best</th>
<th>Worst</th>
<th>Average</th>
<th>Standard deviation</th>
<th>Coefficient of variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_a$</td>
<td>$-0.7011$</td>
<td>$-41.2833^a$</td>
<td>$-20.8734^a$</td>
<td>$24.8948$</td>
<td>$-1.1927$</td>
</tr>
<tr>
<td>$Z_q$</td>
<td>$0.2307$</td>
<td>$-7.6899^a$</td>
<td>$6.4499^a$</td>
<td>$21.3800$</td>
<td>$3.3148$</td>
</tr>
<tr>
<td>$M_q$</td>
<td>$-1.1758$</td>
<td>$0.6724^a$</td>
<td>$1.2852^a$</td>
<td>$5.3128$</td>
<td>$4.1339$</td>
</tr>
<tr>
<td>$Z_{be}$</td>
<td>$0.3840$</td>
<td>$11.8288^a$</td>
<td>$2.8878$</td>
<td>$30.0194$</td>
<td>$10.3952$</td>
</tr>
<tr>
<td>$M_{be}$</td>
<td>$-7.1646$</td>
<td>$8.0484^a$</td>
<td>$0.0020^a$</td>
<td>$35.1770$</td>
<td>$1.8033 \times 10^4$</td>
</tr>
<tr>
<td>$b_{x1}$</td>
<td>$-0.0093$</td>
<td>$-5.3487^a$</td>
<td>$-1.2317$</td>
<td>$3.5939$</td>
<td>$-2.9179$</td>
</tr>
<tr>
<td>$b_{x2}$</td>
<td>$0.2037$</td>
<td>$35.4240^a$</td>
<td>$1.8770^a$</td>
<td>$9.2784$</td>
<td>$4.9432$</td>
</tr>
<tr>
<td>$b_{y1}$</td>
<td>$0.0774$</td>
<td>$0.0892$</td>
<td>$2.7547^a$</td>
<td>$18.7339$</td>
<td>$6.8007$</td>
</tr>
<tr>
<td>$b_{y2}$</td>
<td>$0.0182$</td>
<td>$0.0162$</td>
<td>$2.6264^a$</td>
<td>$7.1595$</td>
<td>$2.7260$</td>
</tr>
<tr>
<td>$J$</td>
<td>$1.1745 \times 10^{-11}$</td>
<td>$1.1962 \times 10^{151}$</td>
<td>$7.4764 \times 10^{149}$</td>
<td>$2.9905 \times 10^{150}$</td>
<td>$4$</td>
</tr>
</tbody>
</table>

$^a$Boundary violation.

Table 8. Identification results for LC-LM (strategy 3).

<table>
<thead>
<tr>
<th></th>
<th>Best</th>
<th>Worst</th>
<th>Average</th>
<th>Standard deviation</th>
<th>Coefficient of variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_a$</td>
<td>$-0.7012$</td>
<td>$-0.7013$</td>
<td>$-0.7012$</td>
<td>$0.0504 \times 10^{-3}$</td>
<td>$-0.0718 \times 10^{-3}$</td>
</tr>
<tr>
<td>$Z_q$</td>
<td>$0.2308$</td>
<td>$0.2308$</td>
<td>$0.2308$</td>
<td>$0.0235 \times 10^{-3}$</td>
<td>$0.1019 \times 10^{-3}$</td>
</tr>
<tr>
<td>$M_a$</td>
<td>$-2.3688$</td>
<td>$-2.3700$</td>
<td>$-2.3689$</td>
<td>$0.5126 \times 10^{-3}$</td>
<td>$-0.2164 \times 10^{-3}$</td>
</tr>
<tr>
<td>$M_q$</td>
<td>$-1.1760$</td>
<td>$-1.1754$</td>
<td>$-1.1760$</td>
<td>$0.2443 \times 10^{-3}$</td>
<td>$-0.2077 \times 10^{-3}$</td>
</tr>
<tr>
<td>$Z_{be}$</td>
<td>$0.3841$</td>
<td>$0.3842$</td>
<td>$0.3841$</td>
<td>$0.0795 \times 10^{-3}$</td>
<td>$0.2069 \times 10^{-3}$</td>
</tr>
<tr>
<td>$M_{be}$</td>
<td>$-7.1653$</td>
<td>$-7.1634$</td>
<td>$-7.1652$</td>
<td>$0.8164 \times 10^{-3}$</td>
<td>$-0.1139 \times 10^{-3}$</td>
</tr>
<tr>
<td>$b_{x1}$</td>
<td>$0.0093$</td>
<td>$0.0093$</td>
<td>$0.0093$</td>
<td>$0.0025 \times 10^{-3}$</td>
<td>$-0.2640 \times 10^{-3}$</td>
</tr>
<tr>
<td>$b_{x2}$</td>
<td>$0.0237$</td>
<td>$0.0237$</td>
<td>$0.0237$</td>
<td>$0.0108 \times 10^{-3}$</td>
<td>$0.0530 \times 10^{-3}$</td>
</tr>
<tr>
<td>$b_{y1}$</td>
<td>$0.0774$</td>
<td>$0.0774$</td>
<td>$0.0774$</td>
<td>$0.0037 \times 10^{-3}$</td>
<td>$0.0472 \times 10^{-3}$</td>
</tr>
<tr>
<td>$b_{y2}$</td>
<td>$0.0182$</td>
<td>$0.0182$</td>
<td>$0.0182$</td>
<td>$0.0016 \times 10^{-3}$</td>
<td>$0.0854 \times 10^{-3}$</td>
</tr>
<tr>
<td>$J$</td>
<td>$1.1745 \times 10^{-11}$</td>
<td>$1.1746 \times 10^{-11}$</td>
<td>$1.1745 \times 10^{-11}$</td>
<td>$2.6429 \times 10^{-16}$</td>
<td>$2.2501 \times 10^{-5}$</td>
</tr>
</tbody>
</table>
Figure 7(a) show that the results found by LC are relatively close to each other, mainly considering the random nature of this algorithm. Strategy 2 was the worst choice by far. From the set of 20 runs, two ended with an indetermination for the value of the functional $J$, and the other two runs ended with $J = +\infty$ and $J = -\infty$. These facts can have been caused by a numerical ill-conditioning. Consequently, Table 7 was built computing only the remaining 16 runs. It can be seen that for the worst results and for the average of the results as well, there are some boundary violations. At the end, these results were very poor revealing that LM has a strong dependency on the initial guess for finding good results, as illustrated in see Figure 7(b). Finally, Table 8 shows that if properly initialized, LM is capable of ending the local optimization and thus finding solutions extremely close to each other.

Figures 8 and 9 show the performance and convergence of both strategies 1 and 3 for the better and the worse case of each one. They provide a comparison basis to analyse the behaviour of the used techniques.
Figure 8. LC evolution and performance. (a) LC self-adaptation (best case). (b) LC self-adaptation (worst case). (c) LC stage’s evolution (best case). (d) LC stage’s evolution (worst case). (e) Angle of attack (best case). (f) Angle of attack (worst case). (g) Pitch angular rate (best case). (h) Pitch angular rate (worst case).
For strategy 1, the evolution and results of the LC can be observed in Figure 8. Figure 8(a) and (b) demonstrate the transitions due to its self-adaptation skills. Figure 8(c) and (d) illustrate which heuristics is conducting the optimization process at a given iteration. Figure 8(e) and (f) give the results for the angle of attack. Finally, Figure 8(g)
and (h) show the results for the pitch angular rate. In a certain manner, Figure 8(e)–(h) help to show that even though relatively close, the results found are different one from the other and also they are not good enough to be considered as a final answer for the identification problem.

For strategy 3, the evolution and results of the LM can be observed in Figure 9. Figure 9(a) and (b) demonstrate the evolution of the algorithm along the optimization task. Figure 9(c) and (d) illustrate the results for the angle of attack. Finally, Figure 9(e) and (f) give the results for the pitch angular rate. In opposition to what was viewed for strategy 1, Figure 9(c)–(f) clarify that LC is a good way to obtain proper initial designs for LM. In both cases illustrated in Figure 9, the final answers can be considered precise enough for the identification of the problem.

9. Conclusion

This article presented an identification procedure to determine the longitudinal stability and control derivatives of a training jet aircraft. Within the framework of the output error method, the tested optimization algorithms are based on LC and LM methods. In the present contribution, GA and PSO represented the phases of the LC algorithm. The individual performances of both the algorithms were tested and then a cascade-type scheme was proposed, aimed at taking advantage of the global search capabilities of LC and the local search capabilities of LM. Finally, the experimental investigation illustrated the possibility of using the proposed technique in a real world environment. The results are very encouraging in the sense that more complex models, embracing non-linearities, will be analysed in further research.

Acknowledgements

The authors are thankful to the Brazilian Research Agencies, CNPq, and FAPESP, for the PhD and research scholarships granted to some of the authors. Dr Valder Steffen, Jr acknowledges CNPq for the partial financing (Proc. 470346/2006-0) of this research work. Dr Luiz Carlos Sandoval Góes also acknowledges FAPESP (Proc. 01/08753-0) and CNPq (Proc. 312116/2006-3 and 485806/2006-1) for funding part of this research.

References


