Aircraft Dynamics Identification for Optimal Control

Cédric Rommel, Joseph Frédéric Bonnans, Baptiste Gregorutti, Pierre Martinon

Aircraft Dynamics Identification for Optimal Control

J. F. Bonnans[1,2], B. Gregorutti[3], P. Martinon[1,2] and C. Rommel[1,2,3]
INRIA[1], CMAP[2], Safety Line[3]
cedric.rommel@safety-line.fr · frederic.bonnans@inria.fr
baptiste.gregorutti@safety-line.fr · pierre.martinon@inria.fr

Abstract

Four new Maximum Likelihood based approaches for aircraft dynamics identification are presented and compared. The motivation is the need of accurate dynamic models for minimizing aircraft fuel consumption using optimal control techniques. A robust method for building aerodynamic models is also suggested. All these approaches were validated using real flight data from 25 different aircraft.

1 Introduction

Aircraft dynamics identification has been a longstanding problem in aircraft engineering, and is essential today for the optimization of flight trajectories in aircraft operations. This motivates the search for accurate dynamical systems identification techniques, the main topic of this paper. The application we are most interested in here is aircraft fuel consumption reduction. It is known that this is as a major goal for airlines nowadays, mainly for economic reasons, but also because it implies less CO_2 emissions. We limit our study to civil flights, and more specifically to the climb phase, where we expect to have more room for improvement. The techniques presented hereafter are suited for data extracted from the Quick Access Recorder (QAR). They contain multiple variables such as the pressure altitude and the true airspeed, with a sample rate of one second. According to the literature [8, 11], two widely used approaches for aircraft dynamics estimation are the Output-Error Method and Filter-Error Method, based on the main ideas of measurement error minimization and state dynamics re-estimation. Recent advances include using neural networks for the state estimation part [14]. On the other hand, renewed interest for the older Equation-Error Method has also been observed [12]. We propose in this paper variations of the latter. Adopting a statistical point of view, we state several regression formulations of our problem and solve them using Maximum Likelihood based techniques. We illustrate our methods with numerical results based on real data from 10 471 flights.

Organization of the paper: In section 2 we give a more detailed description of the dynamics identification problem. The parametric models used in our

analysis are presented in section 3 and the estimation approaches tested are described in section 4. Finally, section 5 contains the numerical results obtained.

2 Problem description

We consider an aircraft in climb phase, modeled as a dynamical system of state variables $\boldsymbol{x} = (h, V, \gamma, m) \in \mathbb{X}$ and control variables $\boldsymbol{u} = (\alpha, N_1) \in \mathbb{U}$ (see table 1 for the notations), where $\mathbb{U} = L^\infty(0, t_f; \mathbb{R}^2)$ and $\mathbb{X} = W^{1,\infty}(0, t_f; \mathbb{R}^4)$ (i.e. the set of primitives of functions in $L^\infty(0, t_f; \mathbb{R}^4)$). In such a context, the problem of optimizing the trajectory during a certain horizon $t_f > 0$ may be seen as a nonlinear constrained optimal control problem of the following form:

$$\begin{aligned} &\min_{(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{X} \times \mathbb{U}} \int_0^{t_f} \ell(t, \boldsymbol{u}, \boldsymbol{x}) dt + \varphi(\boldsymbol{x}(0), \boldsymbol{x}(t_f)), \\ &\text{s.t.} \begin{cases} \dot{\boldsymbol{x}} & = g(t, \boldsymbol{u}, \boldsymbol{x}), \quad \text{for a.e. } t \in [0, t_f]; \\ \boldsymbol{u}(t) & \in U_{ad}, \quad \text{for a.e. } t \in [0, t_f], \\ \Phi(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) & \in K_\Phi, \\ c_j(t, \boldsymbol{x}(t)) & \leq 0, \quad j = 1, \ldots, n_c, \quad \text{for all } t \in [0, t_f]. \end{cases} \end{aligned} \tag{1}$$

All the mappings from (1), i.e. the *running cost* ℓ, the *final cost* φ, the *state equation* function g, the *initial-final state constraint* function Φ and *state path constraints* c_j for $j = 1, \ldots, n_c$, are C^∞. The set U_{ad} is assumed to be a closed subset of $\mathbb{R}^2$, K_Φ is assumed to be a nonempty closed convex set of $\mathbb{R}^{n_\Phi}$ and $n_\Phi, n_c \in \mathbb{N}$.

Table 1: Variables nomenclature

Notation	**Meaning**
h	Aircraft altitude
V	Aircraft true airspeed (TAS)
γ	Path angle
m	Aircraft mass
α	Angle of attack (AOA)
N_1	Engines turbofan speed
ρ	Air density
M	Aircraft Mach number
SAT	Static air temperature
θ	Pitch angle
C	Total fuel flow
T	Total thrust force
D, L	Drag and lift forces
C_{sp}	Specific consumption
ψ	Heading angle
μ	Bank angle (around TAS)
W, ξ	Wind speed norm and heading angle
W_x, W_y, W_z	Horizontal and vertical wind speed components

The function g from (1) reflects the fact that we want the optimized trajectory to respect some model of the aircraft dynamics. A model of flight mechanics which can be used for this purpose is:

$$\begin{cases} \dot{h} = & V\sin\gamma + W_z, & (2)\\ \dot{V} = & \dfrac{T\cos\alpha - D - mg\sin\gamma - m\dot{W}_{xv}}{m}, & (3)\\ \dot{\gamma} = & \dfrac{(T\sin\alpha + L)\cos\mu - mg\cos\gamma - m\dot{W}_{zv}}{mV}, & (4)\\ \dot{m} = & -C_{sp}T, & (5) \end{cases}$$

where

$$\begin{cases} \dot{W}_{xv} = \dot{W}_x\cos\psi\cos\gamma + \dot{W}_y\sin\psi\cos\gamma + \dot{W}_z\sin\gamma,\\ \dot{W}_{yv} = -\dot{W}_x\sin\psi + \dot{W}_y\cos\psi,\\ \dot{W}_{zv} = -\dot{W}_x\cos\psi\sin\gamma - \dot{W}_y\sin\psi\sin\gamma + \dot{W}_z\cos\gamma. \end{cases} \quad (6)$$

In the previous system, W_x, W_y, W_z denote the horizontal and vertical wind speeds in the ground frame of reference and μ, ψ the bank and heading angles (see notations in table 1). Such dynamics account for variable wind during the flight as well as turning maneuvers. Several possibilities can emerge from equations (2)-(5), by setting μ, ψ and/or W_x, W_y, W_z to zero. In this article we shall compare two of them: one obtained when $\mu = W_z = W_y = W_x = \psi = 0$, called hereafter *no-wind-dynamics*, and the other one when just $\mu = W_z = 0$, called *wind-dynamics*. Note that the models presented in section 3 stay the same for both dynamics, as well as all identification processes described in section 4. In system (2)-(5), the elements T, D, L and C_{sp} (see table 1 for notations) are unknown and assumed to be functions of the state and control variables. It is quite common in flight mechanics to assume the following dependencies for them, based on physical considerations:

$$\begin{cases} T & \text{function of } M, \rho, N_1,\\ D & \text{function of } M, \rho, \alpha, V,\\ L & \text{function of } M, \rho, \alpha, V,\\ C_{sp} & \text{function of } M, h, SAT. \end{cases} \quad (7)$$

More details on these dependencies and their origins are discussed in section 3.1. Now, given an aircraft for which a sufficient amount of flight data is available, we aim to identify with the highest precision possible some model of its dynamics, which is equivalent to estimate the four functions from (7). We insist that our intention is to get customized dynamics for each individual aircraft, as opposed to one general model for a whole class of airplanes. Among many different variables recorded in the QAR, nine of them seem sufficient for what we intend to do: $h, M, C, N_1, SAT, \theta, \psi, W$ and ξ (see table 1 for notations). All other variables needed are derived from these ones using physical models of common use in flight mechanics and presented in appendix A and B.

3 Parametric models

3.1 Models structure

As stated in paragraph 2, the thrust T, the specific consumption C_{sp}, the drag D and the lift L are assumed to be functions of some physical variables listed in (7). In order to apply parametric identification methods to infer such functions, we need to assume that they belong to some parametrized function space. This is equivalent to choosing the form of the models. For the sake of simplicity, we choose to use linear identification models. In the case of the thrust and the specific consumption, our models were inspired by [15, Chapter 2.2.1]:

$$T(M, \rho, N_1) = N_1 \rho^{0.6} (\boldsymbol{\theta}_1^T M^3 + \boldsymbol{\theta}_2^T), \tag{8}$$

$$C_{sp}(M, h, SAT) = \boldsymbol{\theta}_1^{csp} h + SAT^{\frac{1}{2}} (\boldsymbol{\theta}_2^{csp} + \boldsymbol{\theta}_3^{csp} h + \boldsymbol{\theta}_4^{csp} M + \boldsymbol{\theta}_5^{csp} hM), \tag{9}$$

where $\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \dots \boldsymbol{\theta}_5^{csp}$ are parameters to be estimated. Models (8)-(9) may be rewritten so to make linearity explicit:

$$T = X_T \cdot \boldsymbol{\theta}_T, \qquad \text{with} \quad \begin{cases} X_T &= N_1 \rho^{0.6} [M^3, 1]^\top, \\ \boldsymbol{\theta}_T &= [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T]^\top, \end{cases} \tag{10}$$

$$C_{sp} = X_{csp} \cdot \boldsymbol{\theta}_{csp}, \qquad \text{with} \quad \begin{cases} X_{csp} &= [h, SAT^{\frac{1}{2}}, SAT^{\frac{1}{2}} h, SAT^{\frac{1}{2}} M, SAT^{\frac{1}{2}} hM]^\top, \\ \boldsymbol{\theta}_{csp} &= [\boldsymbol{\theta}_1^{csp}, \dots, \boldsymbol{\theta}_5^{csp}]^\top. \end{cases} \tag{11}$$

Concerning the aerodynamic forces, we use the common model

$$\begin{cases} L(M, \rho, V, \alpha) &= \frac{1}{2} \rho V^2 S C_z(\alpha, M), \\ D(M, \rho, V, \alpha) &= \frac{1}{2} \rho V^2 S C_x(\alpha, M), \end{cases} \tag{12}$$

where S is the wing planform area, C_x is the drag coefficient and C_z is the lift coefficient. Ignoring the value of S, we model the products SC_x and SC_z as affine functions of a family of monomials derived from the couple of variables (α, M). The technique used to select these monomials is described in the next paragraph.

3.2 Aerodynamic features selection

Let $d \in \mathbb{N}$ such that $d > 1$. We consider a family of monomials of degree at most d of the couple of variables (α, M):

$$X = (X_1, \dots, X_r) = \left(\alpha^k M^{j-k} : j = 0, \dots, d \quad ; \quad k = 0, \dots, j\right) = (1, M, \alpha, M\alpha, \dots), \tag{13}$$

where $r = \begin{pmatrix} d+2 \\ 2 \end{pmatrix}$. We assume hereafter that SC_x and SC_z are linear on X:

$$\begin{cases} Y_x = SC_x(\alpha, M) = X \cdot \boldsymbol{\theta}_D + \varepsilon_x, \\ Y_z = SC_z(\alpha, M) = X \cdot \boldsymbol{\theta}_L + \varepsilon_z, \end{cases} \tag{14}$$

where $\cdot$ denotes the standard dot product in Euclidean spaces, $\boldsymbol{\theta}_D, \boldsymbol{\theta}_L$ denote the vectors of parameters and $\varepsilon_x, \varepsilon_z$ are error terms. Let us assume now that $X, Y_x, Y_z, \varepsilon_x$ and ε_z are random variables. We have access to some observations of X: $\{x_1, \dots, x_N\}$. Assuming C_{sp} to be a known function of M, h, SAT (see section 4.1), we are able to reconstruct observations of Y_x and Y_z from the measured data, denoted respectively $\{y_i^x\}_{i=1}^N$ and $\{y_i^z\}_{i=1}^N$.

Performing the estimation of the parameters using all monomials through linear least-squares could lead to *overfitting* (see e.g. [7, Chap. 7.2]). Therefore, we search for a sparse structure of the parameter vectors $\boldsymbol{\theta}_D, \boldsymbol{\theta}_L$, which is commonly known in statistics and machine learning as *feature selection*.

Given the linearity of model (14), a very popular feature selection method is the *Lasso* [18]. It consists in using all possible variables to fit the available data through least-squares penalized by the $L1$-norm of the parameters vector:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|y_i - x_i \cdot \boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1, \tag{15}$$

where $\boldsymbol{\theta}$ denotes either $\boldsymbol{\theta}_D$ or $\boldsymbol{\theta}_L$ and y_i denotes either y_i^x or y_i^z for all $i = 1, \dots, N$. This approach is known to be inconsistent when some of the variables are highly correlated. Indeed, in this case, the variable selection is very sensitive to the training data used. As some of the monomials in X are likely to be correlated, we decided to use an adaptation of the Lasso which is supposed to be consistent under such conditions: the *Bolasso* [2].

The idea of such feature selection technique is to perform the Lasso repeatedly over several *bootstrap replications* of the initial data set, i.e. samples of size N drawn with replacement using a uniform distribution. The selected variables are then given by the intersection of the variables selected over all Lasso executions. This method has been proved to select the right variables with very high probability under the following assumptions (believed to be verified here):

1. the cumulant generating functions $\mathbb{E}\left[\exp(s\|X\|_2^2)\right]$ and $\mathbb{E}\left[\exp(sY^2)\right]$ are finite for some $s > 0$ (where Y denotes either Y_x or Y_z),
2. the joint matrix of second order moments $\mathbb{E}\left[XX^\top\right]$ is invertible,
3. $\mathbb{E}[Y|X] = X \cdot \boldsymbol{\theta}$ and $\mathbb{V}[Y|X] = \sigma^2$ a.s. for some $\boldsymbol{\theta} \in \mathbb{R}^r$ and $\sigma \in \mathbb{R}_+^*$.

The reader is referred to [2, section 3] for further details.

Once the feature selection has been performed for both SC_x and SC_z, we consider vectors containing the selected monomials only, denoted respectively X_{scx} and X_{scz}, of length $p_D, p_L \leq r \in \mathbb{N}^*$. These new variables vectors will be used to determine the values of the parameters $\boldsymbol{\theta}_D \in \mathbb{R}^{p_D}$ and $\boldsymbol{\theta}_L \in \mathbb{R}^{p_L}$, as explained in section 4.

4 Estimation approaches

This section presents four possible identification methods of increasing complexity, all somehow derived from the wide Maximum Likelihood estimator. They will be compared to each other in section 5. For simplicity, all calculations in this section will be based on the *no-wind-dynamics*, but the same approach can be applied to the *wind-dynamics*.

4.1 Single-task Ordinary Least-Squares

The approach presented in this section is the most straightforward one. It will be used as reference for comparison with all other approaches in the following sections. Our first objective is to derive a set of regression problems from system (2)-(5) (where μ, ψ, W_x, W_y, W_z have all been set to 0 for simplicity).

We see that equation (2) does not contain any unknown element to be estimated, which means it is not useful here. While function T appears in (3)-(5), note that D, L and C_{sp} take part in a single equation each. Equation (5) is clearly the problematic one, because of the presence of a product between the unknowns C_{sp} and T. This means that we do not have linearity on the parameters of the r.h.s. of this equations, but also that it cannot be used alone to determine both elements separately: only the product of them is *identifiable* here, in the sense of definition 4.1 (see e.g. [19, Chap 2.6.1]).

Definition 4.1 *Identifiable model*

Let $\mathcal{M} : \Theta \to \mathcal{Y}$ be a model defined on some parameters space Θ into some output space $\mathcal{Y}$. $\mathcal{M}$ is said to be identifiable *if for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta$,*

$$\mathcal{M}(\boldsymbol{\theta}_1) = \mathcal{M}(\boldsymbol{\theta}_2) \Rightarrow \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2. \tag{16}$$

The simplest way of overcoming such difficulties is to choose not to infer C_{sp} and use a general model of it instead. Even though this idea does not solve the real problem, its simplicity led us to test it. For this, we chose the specific consumption model from [15, Chapter 2.1.2], that we shall denote C_{sp}^{ER} hereafter. Thus, in this approach, only T, D and L are to be estimated based on equations (3)-(5).

By isolating only a single unknown term in the right-hand side of each equation, the system may be rewritten as follows:

$$\begin{cases} -m\dot{V} - mg\sin\gamma + \dfrac{C}{C_{sp}^{ER}}\cos\alpha = D, \\ mV\dot{\gamma} + mg\cos\gamma - \dfrac{C}{C_{sp}^{ER}}\sin\alpha = L, \\ \dfrac{C}{C_{sp}^{ER}} = T. \end{cases} \tag{17}$$

In system (17) we make two assumptions: (i) the mass rate $\dot{m}$ has been replaced by the negative total fuel flow $-C$ and (ii) that T is assumed to be strictly equal to C/C_{sp}^{ER}. Substituting the expressions of T, D and L in (10), (12) and (14), we build the following set of regression problems:

$$\begin{cases} Y_D = X_D \cdot \boldsymbol{\theta}_D + \boldsymbol{\theta}_{D0} + \varepsilon_D, \\ Y_L = X_L \cdot \boldsymbol{\theta}_L + \boldsymbol{\theta}_{L0} + \varepsilon_L, \\ Y_T = X_T \cdot \boldsymbol{\theta}_T + \boldsymbol{\theta}_{T0} + \varepsilon_T, \end{cases} \tag{18}$$

with

$$
\begin{aligned}
Y_D &= -m\dot{V} - mg\sin\gamma + \frac{C}{C_{sp}^{ER}}\cos\alpha, \\
Y_L &= mV\dot{\gamma} + mg\cos\gamma - \frac{C}{C_{sp}^{ER}}\sin\alpha, \\
Y_T = \frac{C}{C_{sp}^{ER}}, \qquad X_D &= \frac{1}{2}\rho V^2 X_{scx}, \qquad \text{and} \quad X_L = \frac{1}{2}\rho V^2 X_{scz}.
\end{aligned} \tag{19}
$$

The *intercepts* $\boldsymbol{\theta}_{D0}, \boldsymbol{\theta}_{L0}$ and $\boldsymbol{\theta}_{T0}$, which are also estimated, should account for possible offsets in the dynamics formulation, and $\varepsilon_D, \varepsilon_L, \varepsilon_T$ are random variables representing the noise of in the models.

Remark Had we different assumptions for the dynamics of our aircraft, as described in section 2, this would simply add some wind terms in (17) and (19), but would not change the structure of the problem.

We consider now only one of this regression problems taken separately, indexed by $\ell \in \{D, L, T\}$. Given a sample $\big\{(x_{\ell,1}, y_{\ell,1}), \dots, (x_{\ell,N}, y_{\ell,N})\big\}$ drawn independently, with $N \in \mathbb{N}$, we rewrite the regression problem in its matrix form:

$$
\boldsymbol{Y}_\ell = \boldsymbol{X}_\ell \boldsymbol{\theta}_\ell + \boldsymbol{\varepsilon}_\ell, \tag{20}
$$

with $\boldsymbol{Y}_\ell = [y_{\ell,1}, \dots, y_{\ell,N}]^\top$, $\boldsymbol{\varepsilon}_\ell = [\varepsilon_{\ell,1}, \dots, \varepsilon_{\ell,N}]^\top$ and $\boldsymbol{X}_\ell = [x_{\ell,1}, \dots, x_{\ell,N}]$. Note that each $x_{\ell,i}$ is a column vector of size p_ℓ. Also note that the intercepts have been included in the parameters vectors, by adding a column of ones to the matrices $\boldsymbol{X}_\ell$. We can then solve each of these regression problems separately through *Ordinary Least-Squares* estimators:

$$
\hat{\boldsymbol{\theta}}_\ell = (\boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell)^{-1} \boldsymbol{X}_\ell^\top \boldsymbol{Y}_\ell. \tag{21}
$$

Such estimator is quickly computed even in the case of large samples and, according to the Gauss-Markov Theorem (see e.g. [3, p.215]), it is the *Best Linear Unbiased Estimator*[1] under the following assumptions:

- the covariates observations $\{x_{\ell,i}, i = 1, \dots, N\}$ are known (not drawn from a random variable);
- the noises have zero mean: $\mathbb{E}[\varepsilon_{\ell,i}] = 0$, for $i = 1, \dots, N$;
- they are *homoscedastic*: $\mathbb{V}[\varepsilon_{\ell,i}] = \sigma^2 < \infty$, for $i = 1, \dots, N$;
- and are mutually *uncorrelated*: $\mathrm{Cov}[\varepsilon_{\ell,i}, \varepsilon_{\ell,j}] = 0$, for any $i \neq j$.

Such assumptions are not all verified in practice. We know for example that there is some uncertainty in our covariates observations and that the autocorrelation between data points is quite significant. Considering this, extended versions of this estimator may be better choices in our case, such as the *Generalized Least-Squares* [1] or the *Total Least-Squares* (see e.g. [8, Chap 6.5]). These options, however, would still not allow us to infer C_{sp}, which means we would not be solving the initial problem. Problem (21) is solved using the function LINALG.LSTSQ from Python's SCIPY.LINALG library, which is based on the SVD decomposition of $\boldsymbol{X}_\ell$ (see section 5.2).

[1] "*Best*" means of minimum variance here.

4.2 Multi-task nonlinear least-squares

As stated in section 4.1, two of the main difficulties of the identification problem treated here are:

1. the *non-identifiability* (see definition 4.1) of the product of T and C_{sp} in equation (5) and
2. the fact that T is present in (3)-(5), which means that we have no chance of obtaining the same results three times if we used theses equations separately to identify it.

These obstacles led us to use the three equations together, in a multi-task regression framework [5]. The main idea here is that multi-task learning allows us to enforce all equations to share the same thrust function T, which solves difficulty 2. By doing so, we expect information concerning T gathered while identifying equations (3) and (4) will be somehow transferred to equation (5) during the estimation process, which should help to reduce the non-identifiability issue 1.

Unlike what was done in (17) for the Single-task Ordinary Least-Squares, here we will leave T in the r.h.s. of equations (3)-(5), since we want to use all of them to estimate it:

$$\begin{cases} m\dot{V} + mg\sin\gamma = T\cos\alpha - D, \\ mV\dot{\gamma} + mg\cos\gamma = T\sin\alpha + L, \\ C = C_{sp}T. \end{cases} \tag{22}$$

As before, by injecting T, D, L and C_{sp} as defined in (10)-(12) and (14), we build the set of regression problems:

$$\begin{cases} Y_1 = X_1 \cdot \boldsymbol{\theta}_1 + \varepsilon_1, \\ Y_2 = X_2 \cdot \boldsymbol{\theta}_2 + \varepsilon_2, \\ Y_3 = (X_T \cdot \boldsymbol{\theta}_T)(X_{csp} \cdot \boldsymbol{\theta}_{csp}) + \varepsilon_3, \end{cases} \tag{23}$$

where

$$\begin{aligned} Y_1 &= m\dot{V} + mg\sin\gamma, \\ Y_2 &= mV\dot{\gamma} + mg\cos\gamma, \\ Y_3 &= C, \end{aligned} \tag{24}$$

and

$$X_1 = \begin{bmatrix} X_T\cos\alpha \\ -X_D \end{bmatrix}, \quad X_2 = \begin{bmatrix} X_T\sin\alpha \\ X_L \end{bmatrix}, \quad \boldsymbol{\theta}_1 = \begin{bmatrix} \boldsymbol{\theta}_T \\ \boldsymbol{\theta}_D \end{bmatrix}, \quad \boldsymbol{\theta}_2 = \begin{bmatrix} \boldsymbol{\theta}_T \\ \boldsymbol{\theta}_L \end{bmatrix}. \tag{25}$$

Remark In the third equation of (23), we can still add an *intercept* as in section 4.1, but it cannot be view as an augmented vector X. In this case, we adapt the strategy by fitting the centered targets $\{\tilde{Y}_i = Y_i - \bar{Y}_i\}_{i=1}^3$ without intercepts and setting them *a posteriori* to be equal to the targets means $\{\boldsymbol{\theta}_{i0} = \bar{Y}_i\}_{i=1}^3$.

As previously explained, we will consider the three regression problems from (23) as a single one, called *multi-task regression problem*

$$Y = f(X, \boldsymbol{\theta}) + \varepsilon, \tag{26}$$

where

$$Y = [Y_1, Y_2, Y_3]^\top, \quad \varepsilon = [\varepsilon_1, \varepsilon_2, \varepsilon_3]^\top \in \mathbb{R}^3, \tag{27}$$

$$X = [X_T, X_{csp}, X_L, X_D]^\top \in \mathbb{R}^m, \quad \boldsymbol{\theta} = [\boldsymbol{\theta}_T, \boldsymbol{\theta}_{csp}, \boldsymbol{\theta}_L, \boldsymbol{\theta}_D]^\top \in \mathbb{R}^p. \tag{28}$$

Note that, for any $x \in \mathbb{R}^m$, function $f(x, \cdot) : \boldsymbol{\theta} \mapsto f(X, \boldsymbol{\theta})$ is nonlinear. We will call hereafter a *task* each regression problem from system (23). The number of *tasks* solved by a multi-task regression is hence given by the dimension of the outputs vector Y.

Now let us consider a certain training set $\{(x_i, y_i)\}_{i=1}^N$ drawn from the joint distribution of the random variables X, Y defined in (28) and (27). In order to solve problem (26) we choose here to use the least-squares estimator once again:

$$\hat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta} \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{N} \|y_i - f(x_i, \boldsymbol{\theta})\|_2^2. \tag{29}$$

Problem (29) is an unconstrained non-convex optimization problem. It was solved using the Levenberg-Marquardt algorithm implemented in the LEASTSQ function from Python's SCIPY.OPTIMIZE library (see section 5.2).

4.3 Multi-task maximum likelihood

In this section we will present a different approach than the least-squares for solving the *multi-task regression problem* (26). It differs from the previous one by trying to leverage the correlation between the noises of the different tasks. Let us assume here for more generality that we are dealing with $K \in \mathbb{N}^*$ *tasks*, i.e. $Y, \varepsilon \in \mathbb{R}^K$. Considering a training set $\{(x_i, y_i)\}_{i=1}^N$, we are assuming through our regression model that there is some $\boldsymbol{\theta} \in \mathbb{R}^p$ such that, for any $i = 1, \ldots, N$

$$y_i = f(x_i, \boldsymbol{\theta}) + \varepsilon_i. \tag{30}$$

We assume here that the ε_i are drawn independently from the same distribution $\mathcal{N}(0, \boldsymbol{\Sigma})$ for every $i = 1, \ldots, N$. Then, for a given $\boldsymbol{\theta} \in \mathbb{R}^p$, the samples likelihood writes

$$\mathcal{L}_{ML}(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} [(2\pi)^K \det \boldsymbol{\Sigma}]^{-\frac{1}{2}} \exp\left(-\frac{1}{2} e_i(\boldsymbol{\theta})^\top \boldsymbol{\Sigma}^{-1} e_i(\boldsymbol{\theta})\right), \tag{31}$$

where $e_i(\boldsymbol{\theta}) = y_i - f(\boldsymbol{\theta}, x_i) \in \mathbb{R}^K$ denotes the residue's i^{th} component. The *Log-Likelihood* criterion is in this case

$$\log \mathcal{L}_{ML}(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = -\frac{NK}{2} \log(2\pi) - \frac{N}{2} \log\det \boldsymbol{\Sigma} - \frac{1}{2} \sum_{i=1}^{N} e_i(\boldsymbol{\theta})^\top \boldsymbol{\Sigma}^{-1} e_i(\boldsymbol{\theta}). \tag{32}$$

If we knew the covariance matrix $\boldsymbol{\Sigma}$, the Maximum Likelihood estimator under such assumptions would be obtained by maximizing criterion (32) with

relation to $\boldsymbol{\theta}$, which is equivalent to solving the following *weighted least-squares* problem:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} e_i(\boldsymbol{\theta})^\top \boldsymbol{\Sigma}^{-1} e_i(\boldsymbol{\theta}). \tag{33}$$

Remark It becomes clear through expression (33) that, under the assumption that the K tasks have mutually independent noises of same variance σ (i.e. $\boldsymbol{\Sigma} = \sigma I_K$, with I_K the $K \times K$ Identity matrix), our new estimator is equivalent to the multi-task nonlinear least-squares estimator from section 4.2.

As we do not know $\boldsymbol{\Sigma}$ in our case, we will try to add it as a variable in our optimization problem, which becomes (after eliminating the constant term):

$$\min_{\boldsymbol{\theta},\boldsymbol{\Sigma}} J(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = \frac{N}{2} \log\det \boldsymbol{\Sigma} + \frac{1}{2} \sum_{i=1}^{N} e_i^\top \boldsymbol{\Sigma}^{-1} e_i. \tag{34}$$

First order optimality condition with relation to $\boldsymbol{\Sigma}$ gives $\frac{\partial}{\partial \boldsymbol{\Sigma}} J(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = 0_K$ where 0_K is the $K \times K$ zero matrix and

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} J(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = \frac{N}{2} \frac{\partial}{\partial \boldsymbol{\Sigma}} \left(\log\det \boldsymbol{\Sigma}\right) + \frac{1}{2} \sum_{i=1}^{N} \frac{\partial}{\partial \boldsymbol{\Sigma}} \left(e_i^\top \boldsymbol{\Sigma}^{-1} e_i\right) \tag{35}$$

$$= \frac{N}{2} \boldsymbol{\Sigma}^{-1} - \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1} e_i e_i^\top \boldsymbol{\Sigma}^{-1} = \frac{1}{2} \boldsymbol{\Sigma}^{-1} \left(N I_K - \left[\sum_{i=1}^{N} e_i e_i^\top \right] \boldsymbol{\Sigma}^{-1} \right). \tag{36}$$

See e.g. [6, p.263] for further details. This leads to the classic covariance estimator

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} e_i e_i^\top = \frac{1}{N} \sum_{i=1}^{N} \Big(y_i - f(\boldsymbol{\theta}, x_i)\Big) \Big(y_i - f(\boldsymbol{\theta}, x_i)\Big)^\top. \tag{37}$$

Using such estimator, we may note that

$$\sum_{i=1}^{N} e_i^\top \hat{\boldsymbol{\Sigma}}^{-1} e_i = \operatorname{Tr} \left\{ \left(\sum_{i=1}^{N} e_i e_i^\top \right) \hat{\boldsymbol{\Sigma}}^{-1} \right\} \tag{38}$$

$$= N \operatorname{Tr}(I_K) = NK, \tag{39}$$

which allows us to simplify our log likelihood criterion (34) into

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \log\det \hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}). \tag{40}$$

Problem (40) is unconstrained and non-convex, as (29), but is also highly nonlinear and harder to solve. We solved it using Python's function FMIN_SLSQP from SCIPY.OPTIMIZE library, where $\hat{\boldsymbol{\Sigma}}$ has been constrained to be positive-definite. The following section presents an attempt to facilitate such an optimization task.

4.4 Cholesky maximum likelihood

We chose here to modify criterion $J(\boldsymbol{\theta})$ by computing the modified Cholesky decomposition of $\hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}) = LDL^\top$, where L is a unit lower triangular matrix and D is a positive diagonal matrix. We assume here that $\hat{\boldsymbol{\Sigma}}$ is positive definite. This ensures that $\det \hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta})$ is the product of the diagonal terms of D, which are non-negative:

$$\det \hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}) = \prod_{j=1}^{K} D_{jj}. \tag{41}$$

In practice, we limit the scope of search to a positive definite $\hat{\boldsymbol{\Sigma}}$, which ensures that is no D_{jj} equal to zero.

Hence, optimization problem (40) becomes

$$\begin{aligned} &\min_{\boldsymbol{\theta},L,D} \sum_{j=1}^{K} \log D_{jj} \\ &\text{such that } \begin{cases} D_{jj} > 0, \quad \forall j = 1, \dots, K \\ \hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}) = LDL^\top, \end{cases} \end{aligned} \tag{42}$$

where L and D are possible Cholesky decomposition matrices of $\hat{\boldsymbol{\Sigma}}$. Compared to (40), this new problem is not unconstrained anymore and its unknowns lie in a space of higher dimension. Hence, at first view, our new problem may seem more complex than the former, even though the objective function has been greatly simplified in the process. However, (40) is a reduction of (42) and we believe, based on empirical experience, that reduced problems are often more difficult to solve. This motivated us to use criterion (42). The algorithm chosen to solve it was, once again, FMIN_SLSQP from SCIPY.OPTIMIZE library (see section 5.2). This algorithm, as well as the three previous ones, has a computation time for each iteration which increases linearly with the size of the training sample.

5 Numerical experiments: application to flight data

In this section we will compare all methods presented above on real flight data. The quality criteria chosen will be described in the first subsection. Those are defined for a given data set of $q > 0$ flights.

5.1 Assessment criteria

Let us consider a certain estimator (e.g. one of those presented in section 4) allowing to infer the parameters $\hat{\boldsymbol{\theta}} = [\hat{\boldsymbol{\theta}}_T, \hat{\boldsymbol{\theta}}_{csp}, \hat{\boldsymbol{\theta}}_L, \hat{\boldsymbol{\theta}}_D]^\top$. A leave-one-out based method is defined here. The models are trained on $q-1$ flights (the training flights). Then predictions are made on the remaining flight (the test flight), assumed here to have been recorded on the times $t_0 = 0, \dots, t_n = t_f$. This allows to check if our estimator performs well on new data, which prevents us from considering overfitting as good performance. More precisely predictions of T, C_{sp}, L and D are built thanks to (10)-(12) and (14) and the estimated

parameters $\hat{\boldsymbol{\theta}}$. Based on the chosen dynamics system (as for example (2)-(5)), the states derivatives $\hat{\dot{\boldsymbol{x}}} = [\hat{\dot{h}}, \hat{\dot{V}}, \hat{\dot{\gamma}}, \hat{\dot{m}}]^\top$ can be finally reconstructed. From a regression point of view, we might assess a given estimator by computing the mean squared error of the reconstructed state derivatives (see section 5.2) and estimator prediction for the test flight:

$$\mathcal{C}_1 = \frac{1}{n+1} \sum_{k=0}^{n} \|\dot{\boldsymbol{x}}_{t_k} - \hat{\dot{\boldsymbol{x}}}_{t_k}\|_x^2. \tag{43}$$

Here, $\| \boldsymbol{.} \|_x$ denotes a scaling norm which brings all the components of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ to the same order of magnitude before applying the usual Euclidean norm in $\mathbb{R}^4$. Based on the ideas of M. Stone [16], this criterion can be computed q times, by permuting the selected test flight. Averaging the value of $\mathcal{C}_1$ over all permutations leads to a new performance criterion. Such a technique is well-known in statistics and machine learning as *cross-validation*.

The issue with the previous criterion is that it is static: the fact that the observations from the test flight are time series does not reflect in it. However, as stated in section 2, the final objective of our approaches is to estimate an accurate model of the aircraft dynamics in order to inject it in an optimal control problem (1). For solving such a problem, the identified dynamic system will be integrated several times using different controls in order to look for the admissible control sequence which minimizes the objective function of (1). A good estimator, when reintegrating the flight dynamics with the controls from the recorded flight, should give a trajectory close to the actual one. Noting $(\boldsymbol{x}_m, \boldsymbol{u}_m)$ the measured state and control sequences of a given test flight, and $\hat{\boldsymbol{x}}$ the trajectory reintegrated with $\boldsymbol{u}_m$, we can rate an estimator $\hat{\boldsymbol{\theta}}$ identified using the training flights by looking at the distance between the two state trajectories:

$$\mathcal{C}_2 = \frac{1}{n+1} \sum_{k=0}^{n} \|\boldsymbol{x}_{t_k} - \hat{\boldsymbol{x}}_{t_k}\|_x^2. \tag{44}$$

In practice, we cannot use this approach for the quality assessment. Indeed, the small errors on the estimations of $\hat{\dot{\boldsymbol{x}}}$ accumulate over the integration steps leading systematically to integrated states which move away quite rapidly from the measured states. To counter such phenomena, we try to find, for a given test flight, a control sequence not very far from the measured one which can bring the integrated state sequence as close as possible to the real state sequence. This idea can be written as the following optimization problem:

$$\min_{\boldsymbol{u}} \int_0^{t_f} \left(\|\boldsymbol{u}(t) - \boldsymbol{u}_m(t)\|_u^2 + \|\hat{\boldsymbol{x}}(\boldsymbol{u}, t) - \boldsymbol{x}_m(t)\|_x^2 \right) dt \tag{45}$$

The norm $\| \boldsymbol{.} \|_u$ is an analog of $\| \boldsymbol{.} \|_x$, but scaling $\boldsymbol{u}$ and $\boldsymbol{u}_m$. The mapping $\hat{\boldsymbol{x}} : \mathbb{U} \times \mathbb{R} \to \mathbb{R}^4$ denotes the function used to compute the integrated states with some control function $\boldsymbol{u} \in L^\infty(0, t_f; \mathbb{R}^2)$ and the states derivatives obtained using $\hat{\boldsymbol{\theta}}$. If we denote $\boldsymbol{u}^*$ a solution of problem (45), we can compute the following performance index:

$$\mathcal{C}_3 = \frac{1}{n+1} \sum_{k=0}^{n} \left(\|\boldsymbol{x}_{t_k} - \hat{\boldsymbol{x}}^*_{t_k}\|_x^2 + \|\boldsymbol{u}_{t_k} - \boldsymbol{u}^*_{t_k}\|_u^2 \right), \tag{46}$$

where

$$\boldsymbol{u}^*_{t_k} = \boldsymbol{u}^*(t_k), \qquad \hat{\boldsymbol{x}}^*_{t_k} = \hat{\boldsymbol{x}}(\boldsymbol{u}^*, t_k), \qquad \text{for} \quad t_k = 0, \dots, t_f. \tag{47}$$

As $\mathcal{C}_1$, this last criterion can be cross-validated across all possible permutations of the test flight.

5.2 Implementation details and data description

We compared the identification approaches described in section 4 using real QAR data. All data sets used contained raw measurements of $(h, M, C, N_1, SAT, \theta, \psi, W, \xi)$. These have been smoothed using *smoothing splines* (see Appendix C) and then used to compute *a posteriori* all the other variables needed (see Appendices A and B). In particular, the differentiated variables, such as $\dot{V}$ and $\dot{\gamma}$, were computed analytically using the derivatives of measurements smoothed using splines.

The models used for T, C_{sp}, L and D were those presented in section 3.1. Were compared two different dynamics, which are the *no-wind-dynamics* and the *wind-dynamics*, as defined in section 2. The *feature selection* for the models of L and D was carried out previous to the estimation approaches from section 4, using a data set containing 10 471 flights of 25 different Boeing 737. The maximum degree of the monomials selected was $d = 3$. Only the climb phase of the flights were used here, i.e. data corresponding to altitudes between FL100 = 10 000 ft and the *top of climb* (cruise altitude), specific to each flight. Prior to the application of the Bolasso (described in section 3.2), the data set was split into a *model selection set* - 33% of it - and a training set. The penalization parameter λ in (15) was selected a single time for all the 128 bootstrap replications of the Bolasso using the *model selection set* only. For this, 100 values for λ were compared using the *mean square error* in a 50-fold *cross-validation* scheme (see e.g. [7, Chap. 7.10]). Concerning the estimation part, we wanted to identify a single aircraft (B737). Hence, a subset of $q = 424$ flights was used for this purpose, which corresponds to 334 531 observations (after outliers deletion).

In practice, the Lasso estimations needed for the feature selection part were carried out using the LASSO and LASSOCV functions of Python's SCIKIT-LEARN.LINEAR_MODEL library [13]. The latter are based on a *coordinate descent* algorithm to solve the optimization problem (15). Concerning the optimization part of the estimation approaches, they are all performed using functions of Python's SCIPY library [9], as listed in table 2. As the Linear Least Squares is the only method which is not iterative, its solution was used to initialize the Nonlinear Least Squares, whose solution was used to initialize the other two appoaches. The optimization of problem (45) was performed using BOCOP [17, 4].

Table 2: Optimization algorithms used

Estimation approach	**Function used**	**Algorithm**
Single-task Ordinary Least-Squares	LINALG.LSTSQ	SVD decomposition
Multi-task Nonlinear Least-Squares	OPTIMIZE.LEASTSQ	Levenberg-Marquardt
Multi-task Maximum Likelihood	OPTIMIZE.FMIN_SLSQP	Dieter Kraft's SQP algorithm [10]
Cholesky Maximum Likelihood	OPTIMIZE.FMIN_SLSQP	Dieter Kraft's SQP algorithm [10]

5.3 Results

Figure 1 shows the cross-validated mean square error path of both $L1$-penalty parameters λ used for the aerodynamic features selection described in section 3.2. The values chosen were at the minimum of these curves, i.e. $1.2\ 10^{-1}$ for SC_x and $6.2\ 10^{-5}$ for SC_z.

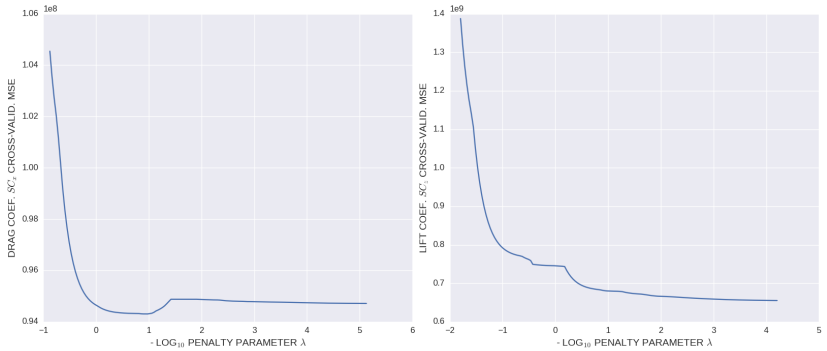
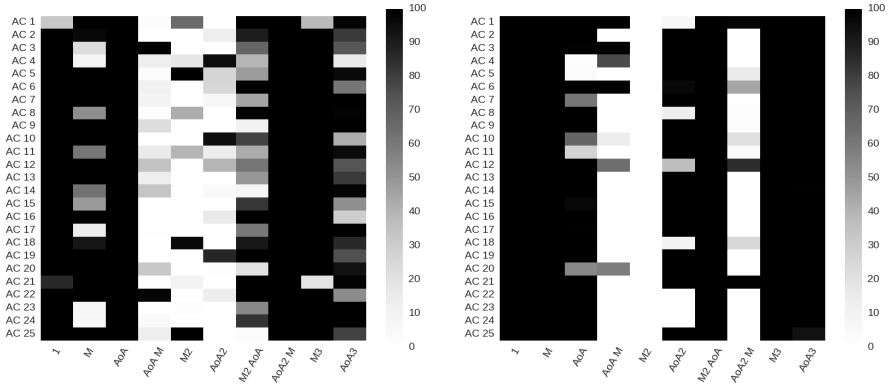


Figure 1: Bolasso penalty parameter selection via cross-validation (no-wind-dynamics)

The results of the feature selection using the *no-wind-dynamics* are presented in figure 2. For both matrices, each row corresponds to a different aircraft and each column to a possible feature to be selected. The color of the cells indicates the selection frequency of given feature for given aircraft among the 128 executions of the Lasso. Columns which are almost entirely black (respectively white) should indicate that the corresponding feature has a high (resp. low) probability of being relevant for the aerodynamic model of the B737. Taking this into account, we see that the lift coefficient features seem easier to select than the drag coefficient ones. Given this results, we chose to keep the following features: $1, M, \alpha, M\alpha^2, M^3$ for the drag coefficient and $1, M, \alpha, \alpha^2, M^2\alpha, M^3, \alpha^3$ for the lift.

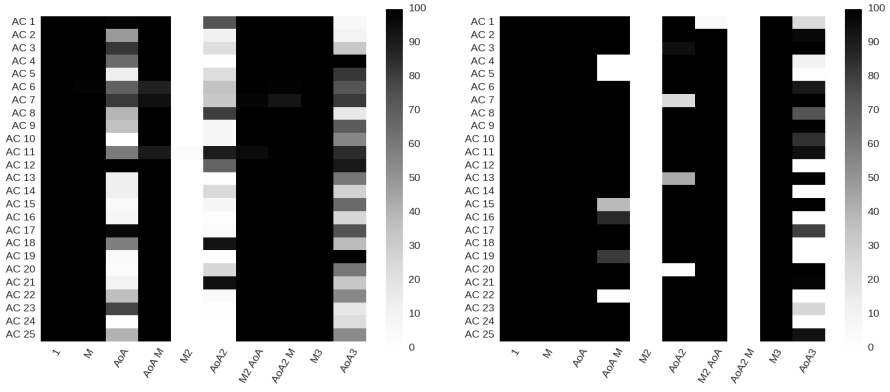
Figure 3 shows the feature selection results obtained using the *wind dynamics*. The matrices obtained in this case present more homogeneous columns than in the previous one, specially for the drag coefficient (the most influenced by the wind). These seem to indicate that the selection has been improved by the use of this more complex dynamics. The aerodynamic features kept in this case were: $1, M, \alpha M, \alpha M^2, \alpha^2 M, M^3$ for the drag coefficient and $1, M, \alpha, \alpha M, \alpha^2, \alpha M^2, M^3$ for the lift coefficient.

Concerning the estimation part, static results for a single flight are presented for all four approaches in figure 4. They were obtained using the *no-wind-dynamics*. The targets (blue full line) correspond to state variables derivatives computed directly from the data. The dashed and dotted lines are the estimated state variables derivatives. As explained in section 5.1, they were all obtained with parameters identified using data from all recorded flights but this one. We observe that our approaches have a little more trouble to estimate the path angle derivative $\dot{\gamma}$, but are nonetheless able to get quite close to all targets. We obtained similar results for other flights, with some quite systematic biases



(a) SC_x Bolasso results (b) SC_z Bolasso results

Figure 2: Aerodynamic features selection results using no-wind-dynamics.



(a) SC_x Bolasso results (b) SC_z Bolasso results

Figure 3: Aerodynamic features selection results with wind-dynamics.

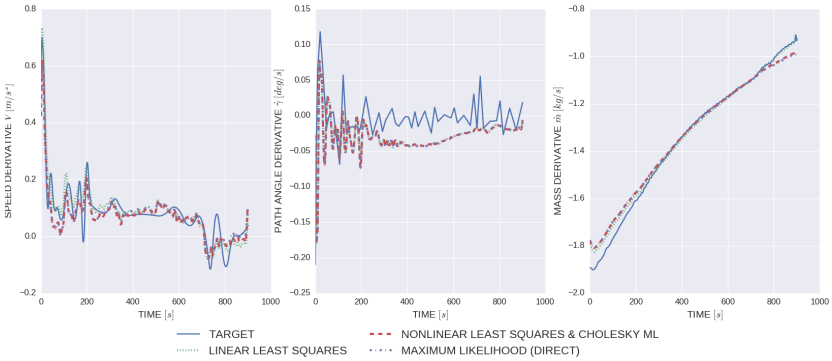


Figure 4: Static results for Linear Least Squares (dotted green), Nonlinear Least Squares (dashed red) and Maximum Likelihood with Cholesky (dashed red) and Direct implementation (dot-dash purple).

such as: the overestimation of $\dot{m}$ by all approaches at the beginning of the flights and the underestimation of it at the end of the flights by the multi-task approaches. Another systematic behavior observed was that the Cholesky Maximum Likelihood approach, which is initialized at the solution given by the Multi-task Nonlinear Least Squares, always stops after a single iteration since it is not able to improve the prediction (reason why there is only on curve for both).

The summary of these static results for all flights with both dynamics are presented in table 3. The latter contains the mean value and standard deviation of the cross-validated criterion $\mathcal{C}_1$ presented in section 5.1. First, we notice that better (lower) mean scores have been obtained using the wind dynamics for all approaches, with no impact on the standard deviation. These results also seem to indicate that the three last multi-task approaches outperform in average the single-task Linear Least Squares method, even though the standard deviations obtained are quite high relatively to the mean scores differences. The best precision was obtained by the direct implementation of the Maximum Likelihood approach. This last remark must be put into perspective, since this estimator is more than 20 times slower to train than all others.

Table 3: Cross-validated $\mathcal{C}_1$ criterion results

Method	**No wind dynamics**		**Wind dynamics**		**Training time*** $[s]$	
	Mean	**Std.**	**Mean**	**Std.**	**Mean**	**Std.**
Linear LS	1.103	0.261	1.101	0.262	0.8	0.1
Nonlinear LS	1.025	0.208	0.989	0.205	32.8	4.4
Multi-task ML	1.023	0.207	0.988	0.204	4.8	0.6
Cholesky ML	1.025	0.208	0.989	0.205	672.0	49.7

*For learning sets of size 298846 ± 154, on 2.5 GHz and 24.6 GB RAM

Dynamic results are presented in figure 5, which shows the solution of prob-

lem (45) for the same flight of figure 4. They were obtained using the *no-wind-dynamics* and allow to compare the single-task Linear Least Squares to the multi-task Nonlinear Least Squares. Notice that the curves are plotted as functions of the altitude, which is possible since the latter is monotonic over the climb phase. We see that the predicted dynamics were able to reproduce quite accurately the recorded flight in this case with a minor correction on the angle-of-attack α and a 20% maximum correction on the engines speed N_1. Similar results were obtained for other flights, which seems promising for the use of such techniques prior to optimal control problems. In addition, we observe once again that the multi-task approach returns slightly better results than the single-task one. This has been confirmed by the results presented in table 4, which contains the mean value and standard deviation over all flights of the cross-validated criterion $\mathcal{C}_3$ from section 5.1.

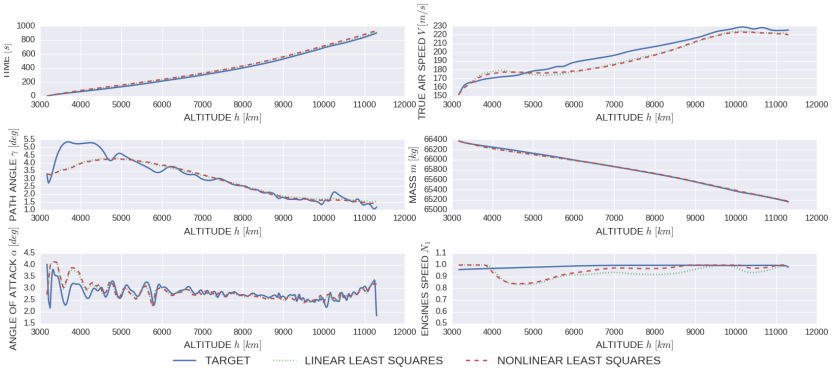


Figure 5: Solution of problem (45) using dynamics estimated by Linear Least Squares (dotted green) and Nonlinear Least Squares (dashed red), in comparison to recorded flight (full blue line).

Table 4: Cross-validated $\mathcal{C}_3$ criterion results, with *no-wind-dynamics*

Method	**Mean**	**Std.**
Linear LS	1.266	0.120
Nonlinear LS	1.245	0.168

6 Conclusion

We presented in this article four different versions of Maximum Likelihood estimators, which can be classified under the broad category of the Equation Error Method [12] for aircraft system identification. A discussion led to the construction of 2 different criteria to assess and compare the performance of such methods: a static criterion and a dynamic one. The results of the static criterion seem to indicate that all the estimators presented here are able to approximate with good precision the dynamics of a given aircraft using its historic

QAR data, including in a *big data* framework. The results from the dynamic criterion seem to indicate that these methods are suited to be used *a priori* to infer the dynamics defining an optimal control problem. The comparison between the four methods showed that the use of a multi-task scheme in three of them led to better accuracy and, in addition, allowed to estimate all unknown functions of the problem: T, L, D and C_{sp}.

Was also presented in this article the Bolasso as a method to infer from the same data the features of linear models of aerodynamic coefficients. The results confirm that this approach is quite robust and can be applied in this context.

Finally, two different possibilities for the dynamic system expression were suggested: one taking into account the wind and aircraft horizontal maneuvers and the other one not. A better performance could be obtained for all estimators and for the aerodynamic features selection using the more complex dynamics, at no additional computational cost.

References

[1] A. C. Aitken. On least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1936.

[2] F. Bach. Bolasso: model consistent Lasso estimation through the bootstrap. pages 33–40. ACM, 2008.

[3] F. Bonnans. *Optimisation Continue*. Dunod,, 2006.

[4] J. F. Bonnans, D. Giorgi, V. Grelard, B. Heymann, S. Maindrault, P. Martinon, O. Tissot, and J. Liu. Bocop – A collection of examples. Technical report, INRIA, 2017.

[5] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[6] G.C. Goodwin and R.L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Developmental Psychology Series. Academic Press, 1977.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

[8] R. V. Jategaonkar. *Flight Vehicle System Identification: A Time Domain Methdology*. AIAA, 2006.

[9] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[10] D. Kraft. *A Software Package for Sequential Quadratic Programming*. DFVLR, 1988.

[11] R. E. Maine and K. W. Iliff. *Application of Parameter Estimation to Aircraft Stability and Control: The Output-error Approach*. NASA, STIB, 1986.

[12] E. A. Morelli. Practical aspects of the equation-error method for aircraft parameter estimation. In *AIAA Atmospheric Flight Mechanics Conference*. AIAA, Aug., 21-24 2006.

[13] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

[14] N. K. Peyada and A. K. Ghosh. Aircraft parameter estimation using a new filtering technique based upon a neural network and Gauss-Newton method. *The Aeronautical Journal*, 113(1142):243–252, 2009.

[15] E. Roux. *Pour une approche analytique de la dynamique du vol.* PhD thesis, SUPAERO, 2005.

[16] M. Stone. Cross-validatory choice and assessment of statistical predictions. *J. of the Royal Stat. Soc.. Series B (Methodological)*, 36(2):111–147, 1974.

[17] Inria Saclay Team Commands. Bocop: an open source toolbox for optimal control, 2017.

[18] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. of the Royal Stat. Soc.*, 58:267–288, 1994.

[19] E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer, 1997.

Appendices

A Atmosphere model

A.1 Air density

The International Standard Atmosphere model gives that $\rho = \frac{P}{R_s SAT}$, where $R_s = 287.053 J.kg^{-1}.K^{-1}$, P is the atmospheric pressure expressed in Pascals and SAT is the *Saturated Air Temperature* in Kelvins.

A.2 TAS, Mach number and sound speed

The Mach number is a function of the SAT and the aircraft relative speed in meters per second V, also called *True Airspeed* (TAS):

$$M = \frac{V}{V_{sound}} = \frac{V}{(\lambda R_s SAT)^{\frac{1}{2}}}, \tag{48}$$

V_{sound} being the atmospheric sound speed in meters per second. Consequently, M can either be seen as a measured variable available in QAR data or as a function of two state variables h and V.

B Flight mechanics model

The path angle γ is the angle between the aircraft speed vector and the horizontal direction. The angle of attack α is the angle between the wings' chord and the relative wind. Here we assume the wings' chord is aligned with the thrust vector and with the aircraft longitudinal axis. The pitch θ is the angle between the longitudinal axis and the horizontal axis. Such definitions and assumptions lead to the following equation linking these three variables: $\theta = \alpha + \gamma$.

C Splines smoothing

All QAR data used to compute the results from section 5 were smoothed using smoothing splines. Consider that we want to smooth some data points $\{(t_i, y_i)\}_{i=1}^N$. We suppose that the t_i's are known, the y_i's are corrupted by some random noise ε of unknown distribution and that a deterministic function f exists such that for all i, $y_i = f(t_i) + \varepsilon_i$. The smoothing splines are defined as the solution of some optimization problem depending on some fixed parameter $\lambda \geq 0$, called *regularization* or *smoothing parameter*. It determines the trade-off between how much curvature is allowed for the solution and how close to the data we want it to be.

As for the Bolasso described in section 3.2, a common solution for choosing the value of λ is to use *K-fold cross-validation*. It consists in randomly partitioning the data set in K subsets of same cardinality. For a given value of λ, at each iteration $k = 1, \ldots, K$, the k^{th} subset is considered to be the test set and the other $K - 1$ subsets form the training set used to compute the smoothing spline estimator $\hat{f}_\lambda^k$. The *cross-validation criterion* for the parameter value λ is obtained by averaging the fitting errors over all the K iterations. Such criterion

is then minimized over a restricted finite set of values of λ. An example of the selection of the smoothing parameter for the speed data of a given flight is showed in figure 6.

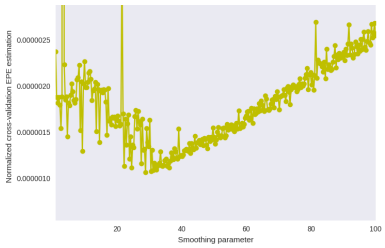


Figure 6: Speed splines smoothing parameter selection via cross-validation