Statistical prediction of aircraft trajectory : regression methods vs point-mass model

Mohammad Ghasemi Hamed, David Gianazza, Mathieu Serrurier, Nicolas Durand

Statistical prediction of aircraft trajectory: regression methods vs point-mass model

M. Ghasemi Hamed[1], D. Gianazza[1,2], M. Serrurier[2], N. Durand[1,2]
[1] *ENAC, MAIAA, F-31055 Toulouse, France*
[2] *Univ. de Toulouse, IRIT, F-31400 Toulouse, France*

***Abstract*—Ground-based aircraft trajectory prediction is a critical issue for air traffic management. A safe and efficient prediction is a prerequisite for the implementation of automated tools that detect and solve conflicts between trajectories. Moreover, regarding the safety constraints, it could be more reasonable to predict intervals rather than precise aircraft positions . In this paper, a standard point-mass model and statistical regression method is used to predict the altitude of climbing aircraft. In addition to the standard linear regression model, two common non-linear regression methods, neural networks and Loess are used. A dataset is extracted from two months of radar and meteorological recordings, and several potential explanatory variables are computed for every sampled climb segment. A Principal Component Analysis allows us to reduce the dimensionality of the problems, using only a subset of principal components as input to the regression methods. The prediction models are scored by performing a 10-fold cross-validation. Statistical regression results method appears promising. The experiment part shows that the proposed regression models are much more efficient than the standard point-mass model. The prediction intervals obtained by our methods have the advantage of being more reliable and narrower than those found by point-mass model.**

Keywords: **aircraft trajectory prediction, point-mass model, BADA, linear regression, neural networks, Loess.**

INTRODUCTION

Predicting aircraft trajectories with great accuracy is central to most operational concepts ([1], [2]) and necessary to the automated tools that are expected to improve the air traffic management (ATM) in the near future. On-board flight management systems predict the aircraft trajectory using a point-mass model of the forces applied to the center of gravity. This model is formulated as a set of differential algebraic equations that must be integrated over a time interval in order to predict the successive aircraft positions in this interval. The point-mass model requires knowledge of the aircraft state (mass, thrust, etc), atmospheric conditions (wind, temperature), and aircraft intent (target speed or climb rate, for example).

Many of this information is not available to ground-based systems, and the available information is not known with good accuracy. The actual aircraft mass is currently not transmitted to the ATM ground systems, although this is being discussed in the EUROCAE group in charge of elaborating the next standards for air-ground datalinks. For a recent reference on the mass estimation problem see [3].

The atmospheric conditions are estimated through meteorological models. Finally, the current ground-based trajectory predictors make fairly basic assumptions on the aircraft intent (see the "airlines procedures" that go with the BADA[1] model). These default "airline procedures" may not reflect the reality, where the target speeds are chosen by the pilots according to a cost index that is a ratio between the cost of operation and the fuel cost. These costs are specific to each airline operator, and not available in the public domain.

As a consequence, ground-based trajectory prediction is currently fairly inaccurate, compared with the on-board prediction. A simple solution would be to downlink the on-board prediction to the ground systems. However, this is not sufficient for all applications: some algorithms ([1], [2], [4], [5], [6], [7]) require the computation of a multitude of alternate trajectories that could not be computed and downlinked fast enough by the on-board predictor. There is a need to compute trajectory predictions in ground systems, for all traffic in a given airspace, with enough speed and accuracy to allow a safe and efficient 4D-trajectory conflict detection and resolution.The literature on trajectory prediction is fairly wide, and one may refer to [8] for a literature survey on the subject, or [9], [10], or [11] for the trajectory predictors' statistical analysis and validation. Other works focus on the benefits provided to ground-based trajectory predictors by using additional, more accurate, input data ([12], [13], [14]). A good introduction on the use of parametric and non-parametric regression methods for trajectory prediction can be found in [15][2]. An interesting model-based stochastic approach is presented in [16], although only validated in a simulation environment.

In this paper, we compare different ways to address the trajectory prediction problem, focusing on the aircraft climb with a 10 minute look-ahead time. We are also interested in finding intervals which contain a desired (i.e. 0.95) proportion of the future aircraft position. Such intervals reflect the prediction uncertainty and can be used for more accurate conflict detection. Climb phase prediction has already been treated by Alligier et al. [17]. Their work addresses the energy rate prediction problem during the climb phase. We selected the climb phase because predicting during this phase is harder and much less accurate than during the cruise phase. As a first approach, the point-mass model is tried with different settings for the model parameters, considering a constant CAS/Mach climb procedure where the aircraft first climbs at

[1]BADA: Base of Aircraft Data
[2]Master's thesis, in french.

a constant Calibrated Air Speed (CAS) until it reaches the CAS/Mach crossover altitude and continues the climb at a constant Mach number. In this approach, the basic parameter setting consists in using the standard CAS and Mach values of the BADA climb procedures file, and a standard reduced thrust during climb, with an average reference aircraft mass. The second setting still uses the reference mass and standard thrust reduction factor, but the actual CAS is computed from the past aircraft positions.

The second approach is radically different and is based on regression methods. The predicted aircraft position is considered as a function $f(x, \beta)$ where x is a vector of input variables and β a vector of parameters. Ghasemi et al. [18] have already applied regression to the trajectory prediction problem. Possibilistic KNN regression [18] consists of predicting intervals rather than precise values. This method focus on finding prediction intervals for K-nearest neighbors (KNN) regression method. For a more generalized application of KNN interval regression based on probability theory the reader can refer to [19].

In this work, the regression input variables are the past aircraft positions, the observed CAS at the current altitude, the deviation of the air temperature from the standard atmosphere, and the predicted wind at different flight levels. The parameters (vector β) must be adjusted using historical data so that the computed output fits the observed position as good as possible. Three well-known regression methods are applied to the trajectory prediction problem. The idea is to see how a parametric linear model, a common parametric non-linear model and an efficient non-parametric model perform on our dataset. In a first attempt, we use a linear regression model to find a linear function $f(\cdot)$ which predicts the altitude z of the aircraft based on the past trajectory. The second model is a feed-forward neural network. Neural networks belong to the class of parametric non-linear models. Finally the Loess method is applied to the trajectory prediction problem. Loess, introduced by Cleveland and Delvin (1988) [20], is a version of locally weighted linear regression which uses K-nearest neighbors as its bandwidth. Locally weighted linear models are very efficient non-parametric estimators[3]. The rest of this paper is organized as follows: section I introduces a widely used simplified point-mass model. Section II introduces regression, and how it is applied to our problem. The three next sections, III, IV, and V, give more details respectively on regression with linear regression, neural networks, locally weighted linear regression. The dataset and experimental setup are detailed in section VI, and results are shown in section VII, before concluding in section VII.

I. The point-mass model

A. Simplified equations

Most ground systems use a simplified point-mass model, sometimes called *total energy model*, to predict aircraft trajectories. This model, illustrated on figure 1, describes the forces applying to the center of gravity of the aircraft and their influence on the aircraft acceleration, making several simplifying assumptions[4]. It is assumed that the thrust and drag vectors are colinear to the airspeed vector, and that the lift is perpendicular to these vectors. Thus, projecting the forces on the airspeed vector axis, the longitudinal acceleration $a = \frac{dV_{\text{TAS}}}{dt}$ along the true airspeed (V_{TAS}) axis can be expressed as follows:

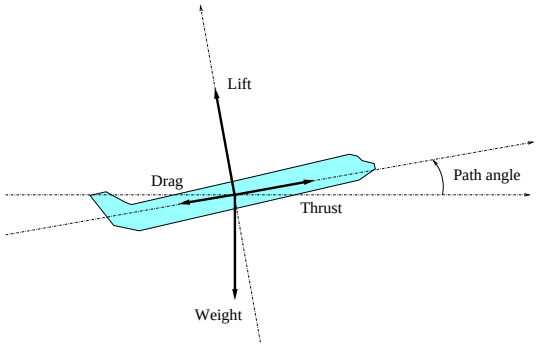

Figure 1. Simplified point-mass model.

$$m.a = T - D - m.g.sin(\gamma) \tag{1}$$

where T is the total thrust, D the aerodynamic drag of the airframe, m the aircraft mass, g the gravitational acceleration, and γ the path angle (i.e. the angle between the airspeed vector and the horizontal plane tangent to the earth surface).

Introducing the rate of climb/descent $\frac{dh}{dt} = V_{\text{TAS}}.sin(\gamma)$, where h is the altitude in meter, this equation can be rewritten as follows (see [22]):

$$(T - D).V_{\text{TAS}} = m.V_{\text{TAS}}.\frac{dV_{\text{TAS}}}{dt} + m.g.\frac{dh}{dt} \tag{2}$$

Sereral equivalent forms of this equation can be used (see Eurocontrol BADA[5] User Manual), depending on what unknown variable is being calculated from the other known variables. Actually using Equation (2) to predict a trajectory requires a model of the aerodynamic drag for any airframe flying at a given speed through the air. In addition, we may need the maximum climb thrust, which depends on what engines the aircraft is equipped with. In the experiments presented here, the Eurocontrol BADA model was used for that purpose.

One cannot use Equation (2) without prior knowledge of the initial state (mass, position, speed,...) of the aircraft, and also of the pilot's intents as to how the aircraft will be operated in the future (thrust law, speed law, or rate of climb). When the aircraft is operated at a given calibrated air speed (CAS[6]) or mach number, computing the true air speed (TAS) requires knowledge of the atmospheric conditions (the air temperature

[3]In section V, we gave a brief overview of their theoretical properties.

[4]Note that more complex point-mass models have been proposed for UAV or fighter airplanes (see [21]), modeling also the side-slip angle.

[5]BADA: Base of Aircraft DAta

[6]CAS: Calibrated Air Speed, which can be assimilated to the speed indicated on the pilot's intruments.

and pressure). Finally, as we need to predict the trajectory over the ground surface, and not only through the air, the wind magnitude and direction are also required.

B. Aircraft operation during climb

Generally, when no external constraint apply during the climb, the aircraft is operated at constant CAS[7] and variable Mach number, until a specified Mach number is reached. Above this CAS/Mach crossover altitude, the aircraft is operated at a constant Mach number, and variable CAS. External constraints may apply, however. After take-off, the aircraft cannot exceed a specified maximum CAS until Flight Level 100[8] is reached. This first climb segment is followed by an acceleration at FL100, and then a second climb segment at a higher calibrated air-speed, until the CAS/Mach crossover altitude is reached.

In this paper, we consider only this second climb segment at constant CAS, followed by the constant Mach climb, as we are mostly interested in predicting the aircraft trajectory in the en-route airspace. Note that some other air traffic control constraints may apply, that modify the aircraft operation during climb. For instance, the aircraft may be operated at a prescribed rate of climb, on some flight segments, in order to be above a specified flight level over a given waypoint.

Even without such constraints, and assuming a climb at constant CAS/Mach, predicting the aircraft trajectory is not easy for ground systems. The actual CAS and Mach values are chosen by the airlines' operators, according to a cost index specific to each airline. The cost index, and the chosen CAS and Mach values are not known by the air traffic control systems , although some studies show the improvements that such knowledge would provide in the trajectory prediction ([14], [12]).

II. Regression methods

In fixed-design regression, there are n pairs of observation $(x_1, Y_1), \cdots, (x_n, Y_n)$, where x_i is the vector of observations known as covariate and Y_i is the response variable. In other words, the random variable Y_i or $Y(x_i)$ follows a mean function $f(x_i)$ with a random error term ε_i defined as:

$$Y_i = f(x_i) + \varepsilon_i, \textit{ where } E(\varepsilon_i) = 0. \tag{3}$$

The model supposes that the ε_i are mutually independents and identically distributed random variables (IID). Thus, the goal is to estimate the mean function $f(\cdot)$ by $\hat{f}(\cdot)$, being is as close as possible to the unknown function $f(\cdot)$. We could also treat the data as random where $(X_1, Y_1), \cdots, (X_n, Y_n)$ are random vectors. In this case, $f(x)$ is interpreted as the mean of Y conditional to $X = x$ as in Equation (4).

$$E(Y|X = x) = f(x). \tag{4}$$

There is little difference in these approaches and in this work, we will take the fixed design approach. The usual assumption is to suppose that the variance of the error is the same everywhere. This is known as homoscedasticity hypothesis and the opposite hypothesis is known as heteroscedasticity.
In least squares regression, the idea is to estimate the mean of $Y(x)$ by $\hat{f}(x)$ and based on some assumptions, it results to finding the function that minimizes the mean of squared errors (MSE), i.e. finding $\hat{f}(\cdot)$ that minimizes :

$$MSE(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 \tag{5}$$

Confidence interval I_β containing a desired proportion β of the response values can be computed by taking inter-quantiles of a normal distribution having its mean and variance, respectively equals to the predicted value $\hat{f}(x)$ and the root mean of squared errors (RMSE) on the training set $MSE(\hat{f})^{\frac{1}{2}}$. For $\beta = 0.95$, we have the following equation :

$$I_{0.95} = [\hat{f}(x) - 1.96 * MSE(\hat{f})^{\frac{1}{2}}, \hat{f}(x) + 1.96 * MSE(\hat{f})^{\frac{1}{2}}] \tag{6}$$

In our trajectory prediction problem, we predict the altitude $z(t)$ at time $t > t_0$, where t_0 is the current time,

The input x is a vector of values extracted from the values:

- The current and previous aircraft states, characterized by $z[k]$, $d[k]$, with $k \in [-10, 0]$. The past trajectory is sampled every δt seconds. $z[k]$ denotes the value measured for the altitude z at time $t = t_0 + k.\delta t$. With this notation, $z[0] = z(t_0)$ is the current altitude, $z[-1]$ is the altitude δt seconds before t_0, and so on. The same notation applies for the distance d;
- The difference between the actual air temperature at sea level and the air temperature of the International Standard Atmosphere (ISA) at sea level;
- The along-track and cross-track wind w and the temperature T at different altitudes;
- Other variables, such as the current *CAS*, *Mach number*, *energy share factor*, *ROCD*, *Ground speed*, etc, and their derivatives with respect to time.

The parameters β must be adjusted using historical data, so that the computed outputs are as close as possible to the observed data. The performance of the tuned model is measured by assessing how the model generalizes on fresh inputs. K-fold cross validation can be used for that purpose.

In order to start with a relatively simple problem, we predict only one future point of the trajectory, N steps ahead. Let us now describe the regression methods that are used to predict the future aircraft position.

III. Linear Regression

Linear regression was the first type of regression analysis to be studied rigorously, and has been used extensively in practical applications. This is due to the fact that regression models which linearly depend on their parameters are easier to fit than non-linear regression models. In statistics, a linear model uses a linear function $f(x)$ to represent the relationship between a dependent random variable Y and a k-dimensional

[7]CAS: Calibrated Air Speed.
[8]FL100 = 10000 feet above isobar 1013 hPa.

vector of predictor variables x. When we have a sample (x_i, y_i)[9] of n observations, in most cases it is not possible to find a linear function $f(\cdot)$ of the k-dimensional input vector x for which Equation $y_i = f(x_i)$ holds for all $i \in (1, \ldots, n)$. So this inequality is modeled through a so called error ε_i which is an unobserved random variable that adds noise to the linear relationship between the dependent variable and regressors. Hence we have :

$$Y_i = f(x_i) + \varepsilon_i = x_i^T \beta + \varepsilon_i,$$

where x_i^T is the transposed vector x_i and β is a p-dimensional ($p = k + 1$) vector of parameters in the linear function $f(\cdot)$. If we stack these n equations together and write the whole in vector form we have :

$$\mathbf{Y} = X\beta + \varepsilon, \tag{7}$$

$$\mathbf{Y} = (Y_1, \cdots, Y_n)^T$$

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{pmatrix}_{(n \times p)}, \varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^T,$$

where $\mathbf{Y}$ is the vector of response variables, $\mathbf{X}$ represents a matrix of all x_i and ε the vector of all errors . In this context, we look for the best estimation of y_i written as $\hat{y}_i$. The response variable is estimated by the equation below in which $\hat{\beta}$ which is an estimation of β :

$$\hat{y}_i = \hat{f}(x_i) = x_i^T \hat{\beta} \tag{8}$$

In the parameter estimation phase, we are searching for the vector of parameters β which fits a straight hyperplane through the set of n points in a way to minimize the sum of squared error function defined by (5). The most common used type of linear regression is the Ordinary Least Square Estimation problem (OLSE) which is described through this section. The assumptions of the OLSE method are stated below :

- The matrix $\mathbf{X}$ must have full column rank p, otherwise we have what is called the multicollinearity in the regressors. Methods for estimating parameters in linear models with multicollinearity have been developed, [23][24] [25] but they require additional assumptions. This assumption holds for our dataset.
- The regressors x_i are assumed to be error-free. It means that they do not have measurement errors and we suppose that this assumption hold for our dataset. The contrary leads to another problem known as errors-in-variables models.
- ε has the following normal distribution :

$$\varepsilon \backsim \mathcal{N}(0, \sigma^2 I). \tag{9}$$

The last statement is one of the most common assumptions in practice and we did similarly. Note that this latter implies also :

[9] y_i is an observation of the random variable Y_i.

- Homoscedasticity : $\forall i, Var[\varepsilon_i^2] = \sigma^2$. The inverse hypothesis, heteroscedasticity, is made when error terms do not have necessarily equal variance. Under such cases it might be better to use a weighted version of the OLSE named Weighted Least Square Estimation (WLSE). WLSE minimizes a weighted version of the sum of squared error terms, where each error term is weighted by a weight that indicates the precision of the information contained in the associated observation. WLSE assumes that the weights are known, which is not our case and this latter forces us to estimate them. Note that when using estimated weights from small numbers of replicated observations, the regression analysis result can be very badly and unpredictably affected. Therefore it is important to use this method when weights can be estimated precisely relative to one another [26]. This is why we made this assumption.
- Non-autocorrelation of errors $E[\varepsilon_i \varepsilon_j] = 0, \forall i \neq j$. In section section VI, we explain that our dataset consists of 1500 pairs of (x_i, Y_i), where the vector x_i has 15 elements and each aircraft trajectory (the pair (x_i, Y_i)) is independent of the other one. This mean that if the aircraft A has an error on its trajectory we can not deduce anything on the error of the aircraft B; so we can assume that the errors are not autocorrelated.

In this situation, the Gauss-Markov theorem states that minimizing the sum of squared residuals gives us the Best Linear Unbiased Estimator (BLUE). In other words, OLSE fits a plane through the set of n vectors in such a way that makes the sum of squared residuals of the model (that is, vertical distances between the points of the data set and the fitted plane) as small as possible.

$$\hat{\beta} = \underset{\beta}{\text{Argmin}}(\textstyle\sum_{i=1}^{n}(y_i - x_i^T\beta)) = \underset{\beta}{\text{Argmin}}(\textstyle\sum_{i=1}^{n}(\varepsilon_i^2)) \ .$$

In OLSE the BLUE estimator is found by :

$$\hat{\beta} = (X^T X)^{-1} X^T y. \tag{10}$$

IV. Regression using neural networks (NN)

Artificial neural networks are algorithms inspired from the biological neurons and synaptic links. An artificial neural network is a graph, with vertices (neurons, or units) and edges (connections) between vertices. There are many types of such networks, associated to a wide range of applications. Beyond the similarities with the biological model, an artificial neural network may be viewed as a statistical processor, making probabilistic assumptions about data ([27]). The reader can refer to [28] and [29] for an extensive presentation of neural networks for pattern recognition. In our experiments, we used a specific class of neural networks, referred to as feed-forward networks, or multi-layer perceptrons (MLP). In such networks, the units (neurons) are arranged in layers, so that all units in successive layers are fully connected. Multi-layers perceptrons have one *input layer*, one or several *hidden layers*, and an *output layer*.

For a network with one hidden layer, the output vector $y = (y_1, ..., y_i, ..., y_n)^T$ is expressed as a function of the input

vector $x = (x_1, ..., x_i, ..., x_k)^T$ as follows:

$$y_k = \Psi(\sum_{j=1}^{n} \beta_{jk} \Phi(\sum_{i=1}^{k} \beta_{ij} x_i + \beta_{0j}) + \beta_{0k}) \qquad (11)$$

where the β_{ij} and β_{jk} are weights assigned to the connections between the input layer and the hidden layer, and between the hidden layer and the output layer, respectively, and where β_{0j} and β_{0k} are biases (or threshold values in the activation of a unit). Φ is an *activation function*, applied to the weighted output of the preceding layer (in that case, the input layer), and Ψ is a function applied by each output unit to the weighted sum of the activations of the hidden layer. This expression can be generalized to networks with several hidden layers.

The output error – *i.e.* the difference between the desired output (target values) and the output y computed by the network – will depend on the parameters β (weights and biases), that must be tuned by training the network, so as to minimize a chosen function of the output error. In our case, the minimized function is the sum of quadratic errors. The optimization method is either a gradient descent with momentum, or a BFGS quasi-Newton method. The activation function is the logistic sigmoïd, and the output function is the identity.

Neural network have already been applied to trajectory prediction, in [30]. However, they were used to predict both the climb and cruise flight segments, given a requested flight level, and lateral navigation as well. This approach, where the altitude error is likely to be small after the cruise flight level has been captured, is difficult to compare to our approach focused on minimizing the prediction error on the climb segment only. A mix of neural networks experts proved efficient for the chosen purpose.

V. Local Polynomial Regression

A. State of the art

Non-parametric regression have been widely studied in 1975-1995, when statistician realized that nonparametric regression could be used for situations where parametric regression methods are not convenient. Several monographs like Eubank (1988) [31], Hastie and Tibshirani (1990) [32], Hardle (1990) [33], Wahba (1990) [34] and Fan and Gijbels (1996) [35] have discussed this topic. The idea of Local Polynomial Regression (LPR) appeared in statistical literature by Stone (1977) [36] and Cleveland (1979) [37]. Cleveland (1979) [37], introduced Locally Weighted Regression (LWR) and a robust version of locally weighted regression known as Robust Locally Weighted regression Scatter plot Smoothing (LOWESS). Cleveland and Delvin (1988) [20] shown that locally weighted linear regression could be very useful in real data modeling applications. They introduced "Loess" which is a multivariate version of locally weighted regression. Their work includes the application of Loess with multivariate predictor dataset and introduction of some statistical procedures analogous to those usually used in parametric regression. They also proposed an ANOVA test for Loess. Fan (1992,1993) [38], [39] studied some theoretical aspects of local polynomial regression. Fan shows that Locally Weighted Linear Regression (LWLR) (or weighted local linear regression) is design-adaptive. It adapts to random and fixed design as seen respectively in Equations (4) and (3). LWLR can be use as well in highly clustered than nearly uniform design. He also shows that the best local linear smoother has 100% efficiency among all possible linear smoothers, including kernel regression, orthogonal series and splines in minimax sense. Another important property of LWLR is their adaptation to boundary points. As shown by Fan and Gijbels (1992) [40], the LWLR estimator does not have boundary effects and therefore it does not require any modifications at the boundary points. This is a very attractive property of these estimators, because in practice, a large proportion of the data can be included in the boundary regions. Then Ruppert and Wand (1994) [41] extend the Fan's results on asymptotic bias and variance to the case of multivariate predictor variables.

B. Locally Weighted Polynomial Regression (LWPR)

LWPR fits a new polynomial for each input instance x in the dataset. The estimation $\hat{y}$ is done by fitting a weighted low degree polynomial model in the neighborhood of the query point x. In general the polynomial degree (d) is 1 or 2. The weight of neighbors of x are found through a kernel function $K(\cdot)$ that gives bigger weights to observations closer to the fitting point x and smaller weights to those farther from x. If we take $d = 0$, LWPR changes into a Kernel regression and if the polynomial is of first degree we have a Locally Weighted Linear Regression (LWLR). The value of the regression function is obtained by evaluating the local polynomial with the x predictor variables values.

1) Definition: suppose that the regression function $f(\cdot)$ at the point x can be approximated locally for z inside a neighborhood of x. The idea is to write the Taylor's expansion for z inside a neighborhood of x as follow [35]:

$$f(z) = \sum_{j=0}^{d} \frac{f^j(x)}{j!}(z-x)^j \equiv \sum_{j=0}^{d} \beta_j (z-x)^j \qquad (12)$$

Equation (12) models the regression function by a polynomial function. Thus for every observation z in neighborhood of x, we write down (12) and we estimate $\beta = (\beta_0, \cdots, \beta_d)^T$ by the vector of parameters $\hat{\beta} = (\hat{\beta}_0, \cdots, \hat{\beta}_d)^T$ which minimizes the locally weighted sum of squares defined in Equation (13), where $K_b(\cdot)$ represents a kernel function with bandwidth b. In fact, estimating $f(x)$ for the random design as well as for the fixed design results in the locally weighted polynomial regression expressed in (13) [35].

$$\sum_{i=1}^{n} K_b(X_i - x)\left(Y_i - \sum_{j=0}^{d} \beta_j (X_i - x)^j\right)^2 \qquad (13)$$

The above formula can be re-expressed as :

$$\sum_{i=1}^{n} w_i \Big(Y_i - \hat{f}(x_i) \Big)^2, \tag{14}$$

where $w_i = K_b(X_i - x)$ *and* $\hat{f}(x_i) = \sum_{j=0}^{d} \beta_j (X_i - x)^j$.

By re-writing (13) in vector notations, we obtain (15)

$$(\mathbf{Y} - \mathbf{X_x}\beta)^T \mathbf{W_x} (\mathbf{Y} - \mathbf{X_x}\beta), \tag{15}$$

where $\mathbf{Y}$ is the vector of response variables and for each x, $\mathbf{X_x}$ *and* $\mathbf{W_x}$ are respectively its predictor matrix and weight matrix as in (16).

$$\mathbf{Y} = (Y_1, \cdots, Y_n)^T$$

$$\mathbf{X_x} = \begin{pmatrix} 1 & (X_1 - x) & \cdots & (X_1 - x)^d \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (X_n - x) & \cdots & (X_n - x)^d \end{pmatrix}_{(n \times (d+1))}, \tag{16}$$

$$\mathbf{W_x} = diag(K(\frac{X_i - x}{b}))_{n \times n}$$

The vector $\hat{\beta}_x$ minimizing this weighted sum of squares is provided by the Weighted Least Square regression:

$$\hat{\beta}_x = ({\mathbf{X_x}}^T \mathbf{W_x} \mathbf{X_x})^{-1} {\mathbf{X_x}}^T \mathbf{W_x} \mathbf{Y} \tag{17}$$

and $\hat{f}(x)$ becomes a linear smoother as in (19).

$$\hat{f}(x) = \sum_{i=1}^{n} a_i(x) Y_i, \tag{18}$$

where $a(x) = I_1^T \hat{\beta}_x$ *and* $I_1^T = (1, 0, \cdots, 0)$.

We can also write the fitted values in vector notation as in Equation (19) where L is the projection matrix or the smoother matrix in which its $L_{ij} = a_j(x_i)$ and $\hat{\mathbf{f}}$ is the vector of fitted values.

$$\hat{\mathbf{f}} = L\mathbf{Y}, \tag{19}$$

$$\hat{\mathbf{f}} = (\hat{f}(x_1), \cdots, \hat{f}(x_n))^T$$

Note that (17) is for single variate regression. When it comes to multivariate LWPR with p predictor variables, the final d columns of $\mathbf{X_x}$ are repeated for each covariate. Hence, $\mathbf{X_x}$ becomes a $n \times (p \times d + 1)$ matrix, $\hat{\beta}_x$ a vector of $(p \times d) + 1$ element and the kernel function a multivariate kernel.

2) Kernel function: In kernel regression or in LWPR , a kernel function $K(\cdot)$ is used to weight the observations. It is chosen so that observations closer to the fitting point x have bigger weights and those farther from x have smaller weights. If K is a kernel, then $K_b(\cdot)$ is also a kernel function.

$$K_b(u) = \frac{1}{b} K(\frac{u}{b}), \ \textit{where } b > 0.$$

Here, b, known as the bandwidth is a constant scalar value used to select an appropriate scale for the data. A kernel function is a non-negative real-valued integrable function K with the properties listed below [37]. Almost all kernel functions respect the first three properties, so they become probability density functions. The last property limits the neighborhood and this helps to achieve better computing performance. For a better explanation of the weight function properties see [37].

$$\forall u, K(-u) = K(u)$$

$$\forall u, K(u) \geq 0$$

$$\int_{-\infty}^{+\infty} K(u)\, du = 1,$$

$$K(u) > 0, |u| < 1.$$

Below, you can see some of the most common kernel choices [42]. Note that $I(\cdot)$ is the indicator function.

- Gaussian : $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$;
- Tricube: $K(u) = \frac{70}{81}(1 - |u|^3)^3 \, \mathbf{I}_{\{|u| \leq 1\}}$;
- Epanechnikov: $K(u) = \frac{3}{4}(1 - u^2) \, \mathbf{I}_{\{|u| \leq 1\}}$;

For multivariate LWPR, the kernel function K_B is a function of p variables. In this case B is a symmetric positive definite $p \times p$ matrix and $|B|$ denotes its determinant. It can be redefined as:

$$K_B(u) = \frac{1}{|B|} K(B^{-1}u). \tag{20}$$

In practice, one can normalize or standardize all the predictors and then use the following kernel :

$$K_b(u) = \frac{1}{b} K(\frac{D(u)}{b}) \tag{21}$$

where $D(\cdot)$ is a distance function like the L_2-norm. Some authors like [37] and [20] took the K-nearest neighbors of x as its neighborhood. In this case, for each x, $b = D_k(x)$ where $D_k(x)$ is the distance from the K-th nearest neighbors (the farthest neighbor) from the query point x. For a detailed discussion on the subject see [43].

C. Loess

Loess introduced by Cleveland and Delvin (1988) [20], is a multivariate version of LOWESS [37]. The weight function chosen by Cleveland and Delvin (1988) [20] was the Tricube kernel, however any other weight function that satisfies the properties listed in the kernel definition could also be used. Loess uses the Euclidean distance to determine distance between points. Let $D_k(x)$ be the Euclidean distance from the K-th nearest neighbors (the farthest neighbor) from the query point x; so $b = D_k(x)$ is the bandwidth for x. The loess weight function is express in Equation (21), where $u = (X_i - x)$, $D(u)$ is u's L_2-norm and $b = D_k(x)$.

D. bandwidth selection in loess

A popular bandwidth selection method is the Leave-One-Out (LOO) technique suggested in Stone (1977)[36] which chooses the following bandwidth b :

$$b = \text{Argmin} \sum_{i=1}^{n} (y_i - \hat{f}'(x)_i)^2, \tag{22}$$

where $\hat{f}'(x)_i$ is the estimation without using the i^{th} observation obtained by the equation 19. Estimating the bandwidth by LOO is a time-consuming task, so it is common to minimize the K-fold cross validation score with $K = 5$ or $K = 10$; this leads to an approximation of LOO. Plug-in bandwidth is another smoothing strategy which is a formula for the asymptotically optimal bandwidth. The plug-in bandwidth requires several unknown quantities that must be estimated from data. In section 4.2 of Fan and Gijbels (1996) [35], a plug-in bandwidth for linear weighted local linear regression is defined. One of the required parameters for this estimator is $f(\cdot)$'s second derivative which is more difficult to estimate than $f(\cdot)$ itself. In this work we use 10-fold cross validation to find the best bandwidth of our dataset.

VI. Data and experimental setup

A. Data pre-processing

Recorded radar tracks from Paris Air Traffic Control Center were used to build the patterns used in the regression methods. This raw data is made of one position report every 1 to 3 seconds, over two months (july 2006, and january 2007). In addition, the wind and temperature data from Meteo France are available at various isobar altitudes over the same two months.

The raw Mode C altitude[10] has a granularity of 100 feet. So the recorded aircraft trajectories were smoothed, using a local quadratic model, in order to obtain: the aircraft position (X,Y in a projection plan, or latitude and longitude in WGS84), the ground velocity vector (V_x, V_y), the smoothed altitude (z, in feet above isobar 1013.25 hPa), the rate of climb or descent (ROCD). The wind (W_x, W_y) and temperature (T) at every trajectory point were interpolated from the meteo datagrid. The temperature at isobar 1000 hPa was also extracted for each point, in order to compute a close approximation of $(\Delta T_0)_{\text{ISA}}$, the difference between the actual temperature and the ISA model temperature at isobar 1013.25 hPa (mean sea level in the ISA atmospheric model). This $(\Delta T_0)_{\text{ISA}}$ is one of the key parameters in the BADA model equations.

Using the position, velocity and wind data, we computed the true air speed (TAS), the distance flown in the air (dAIR), the drift angle, the along-track and cross-track winds (W_{along} and W_{cross}). The successive velocity vectors allowed us to compute the trajectory curvature at each point. The actual aircraft bank angle was then derived from true airspeed and the curvature of the air trajectory. The climb, cruise, and descent segments were identified, using triggers on the rate of climb or descent to detect the transitions between two segments.

Finally, the BADA model equations were used to compute additional data, such as: calibrated airspeed (CAS), Mach number (M), energy share factor[11] (ESF), as well as the derivatives of these quantities with respect to time.

B. Filtering and sampling climb segments

As our aim is to compare several prediction models, we focused on a single aircraft type (Airbus A320), and selected all flights of this type departing from Paris Orly (LFPO) or Paris Roissy-Charles de Gaulle (LFPG). We selected Airbus A320, because this the most common aircraft in Europe. Another technical reason is that introducing other aircraft types forces us to treat the aircraft trajectory prediction problem with more complex models having more parameters and requiring significantly more trajectories. In fact, if we are able to obtain efficient prediction for a single aircraft type, then the investigation of a more complex model, which is a much bigger task, could be easily justified. The trajectories were then filtered so as to keep only the climb segments. An additionnal 40-seconds were clipped from the beginning and end of each segment, so as to remove climb/cruise or cruise/climb transitions. The trajectories were then sampled every 15 seconds, with time and distance origins at the point P_0 where the climb segment crosses flight level FL180[12]. The trajectory segments were sampled so as to obtain 10 points preceding P_0, and a number of points after P_0, depending on the chosen look-ahead time. So the trajectory observed during the preceding time steps (2 minutes 30 seconds), can be used to predict the aircraft position at one or several future time steps. The predicted position can be compared to the actual aircraft position at the same time step.

Trajectories exhibiting a bank angle greater than 5 degrees were discarded, so that the influence of trajectory turns on the rate of climb can be neglected. This allows us to disregard the lateral navigation in our trajectory prediction problem, and focus on the longitudinal and vertical dimensions of the trajectory.

[10]This altitude is directly derived from the air pressure measured by the aircraft. It is the height in feet above isobar 1013.25 hPa.

[11]The energy share factor (ESF) says how much of the energy is devoted to climb or to longitudinal acceleration.

[12]FL180: 18000 feet above isobar 1013 hPa.

C. Building patterns for regression

The regression models $y = f(x, \beta)$ are tuned and assessed using sets of patterns (x, y_d), where x is an input vector, and y_d is the corresponding desired output that can be compared to the computed output y. These patterns, that we have already described in section II were extracted from the sampled climb segments. 1500 patterns were randomly chosen, to build the set used in our experiments.

Each pattern used for regression contains the current ground speed, true and calibrated air speed, Mach number, and their derivatives with respect to time, the energy share factor, the altitude variations and distance flown for the ten preceding time steps, and also the predicted wind and temperature at several altitudes that the aircraft may cross in the look-ahead time. It also contains the potential target variables: distance flown, in the air or above the ground, and altitude reached after N time steps in the future.

D. Principal component analysis

The final patterns set contains 79 numerical variables, measured for 1500 aircraft climbs. There are 76 explanatory variables, and 3 variables to explain (although only the altitude is predicted here). A principal component analysis was performed on the explanatory variables, so as to reduce the dimensonality and avoid redundant input variables in the trajectory prediction. Figure 2 shows the standard deviations of the principal components: 9 components have a standard deviation above 1, and 7 other components are between 0.5 and 1.

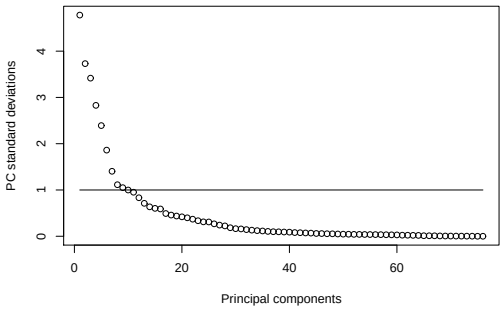

Figure 2. Principal components standard deviations.

Principal components are linear combinations of the initial variables, that we can use as explanatory variables in the regression method. This reduces the dimensionality to 10 to 15 significant principal components, instead of the 76 initial variables. One must keep aware, however, that using linear combinations representing projections on a basis of orthogonal vectors may not take into account some non-linearities in the initial variables.

E. Experimental setup

The methods applied to our prediction problem for climbing aircraft are listed in table I.

BADA	BADA point-mass model, using the reference mass for each aircraft and BADA values for the constant CAS/Mach, assuming reduced climb (eq. 3.8.1 and 3.8.2, p.22 in [22]), and taking account of the $(\Delta T_0)_{\text{ISA}}$ temperature difference.
BADA(obs)	Same BADA model as above, but using the CAS observed at t_0, and the BADA target Mach number
LR	Ordinary least squares linear regression
NN	Regression with neural networks
Loess	Loess

Table I
METHODS APPLIED TO ALTITUDE PREDICTION

In our experiments, these methods are scored using a 10-fold cross-validation on the dataset described in section VI-C. This set is split in ten subsets. Nine of them are used to tune the model parameters β, and the remaining subset is used to assess the model performance. This operation is repeated 10 times, cycling through the subsets. The models' performance is assessed over the ten runs, considering the mean score, the standard deviation, and also confidence intervals for the computed output.

Uncertainty on the prediction can be assessed in the following way: once the model parameters have been tuned on the training set (i.e. the concatenation of 9 subsets used in cross-validation), we can compute a theoritical 95%-confidence interval using the root mean square error (RMSE) observed on this training set, assuming a Gaussian distribution of the error in altitude (and also in distance, if predicted) as explained by Equation (6). We can then count how many instances of the validation subset actually fall within this confidence interval.

VII. RESULTS

Method	MAE	RMSE
BADA	1440 (79)	1824 (95)
BADA(obs)	1440 (77)	1819 (86)
LR	744 (55)	962 (72)
NN	841 (47)	1080 (55)
Loess	699 (54)	908 (72)

Table II
AVERAGE PREDICTION ERRORS (AND STANDARD DEVIATIONS) ON THE ALTITUDE (IN FEET) FOR AIRBUS A320 AIRCRAFT, USING 15 PRINCIPAL COMPONENTS AS INPUT, WITH THE REFERENCE POINT AT FL180 AND A 10-MINUTES LOOK-AHEAD TIME.

Table II shows the prediction errors (mean absolute error, and root mean squared error) over the 10 runs of the cross-validation, for all tested methods. The 15 principal components of higher variance were used as input to the regression methods. This selection was made by prior trials, adding successively the principal components until no significant improvement was observed.

All regression methods perform significantly better than the BADA point-mass model. There are several factors explaining the poor performance of the point-mass models. The parameters'choice assumed a constant CAS/Mach climb at economic thrust, and the same reference mass for all aircraft, which is

actually not the case in reality. Also, the regression methods use the past trajectory to predict the future altitude, whereas our BADA models do not. Using the observed CAS instead of the BADA standard CAS does not improve the results on altitude prediction.

As expected by the theoretical properties of Loess, reviewed briefly before, we can observe that this method gives the best results on our data. As we can see in table II, Loess is more efficient than the linear regression. We have also applied a two-sided Mann-Whitney test (paired Wilcoxon signed rank test) on the errors of Loess and OLSE and the test confirms our statement with a p-vale of 0.0127. It came as a surprise that neural network methods did not perform better than the ordinary least squares linear regression. There may be several explanations to this. Using the principal components as inputs does favour linear methods. In addition, tuning the parameters with the ordinary least squares linear regression can be done with an exact method, whereas neural networks methods require iterative approximations or a stochastic selection process, that may have difficulties to find the optimum when using input variables that are not very efficient in explaining the target variable(s). In fact neural networks, being non-linear estimators, have less bias but bigger variance than OLSE. Due to the high dimensionality of our dataset and the relatively small number of observations, the neural network estimations suffer from high variance. This latter leads the neural networks MSE to be greater than the OLSE's one. For more detail on the bias-variance trade-off and model complexity in regression see [44].

Method	Ratio in theoretical 95% interval	Theoretical 95% $\lvert\delta z\rvert$
BADA	0.92 (0.025)	3279
BADA(obs)	0.93 (0.021)	3369
LR	0.94 (0.013)	1863
NN	0.905 (0.030)	1846
Loess	0.948 (0.022)	1842

Table III
UNCERTAINTY ON THE ALTITUDE PREDICTION (AIRBUS A320), FOR A REFERENCE POINT AT FL180 AND A 10-MINUTES LOOK-AHEAD TIME.

Some results on the uncertainty of the altitude prediction are shown in Table III. The second column shows the ratio of predictions, computed with instances from the validation set, that actually fall within the 95% confidence interval computed using the training set. The third column shows the width of this theoretical interval. We can observe that point-mass models give really wider intervals than the regression models. OLSE and, particularly, the loess model provide reliable intervals having much wider intervals than the BADA models. It is important to note that these intervals are computed on the climb phase which is really hard to estimate.

CONCLUSION

In this paper, we have applied several methods to the prediction of altitude. The aim was to compare these methods when predicting the altitude of climbing aircraft 10 minutes ahead, starting from an initial point at flight level FL180, and possibly using the past trajectory to improve the prediction. Radar and Meteo data recorded over two months (july 2006, january 2007) were used to build a dataset of explanatory and target variables. A principal component analysis of this data allowed us to reduce the dimensionality to 10 to 15 significant components, instead of the 76 initial explanatory variables. The models are compared by performing a 10-fold cross-validation on a set of 1500 climb segments.

Our results show that the regression methods perform significantly better than the point-mass model. This is not surprising as the former learn from the observation of the past trajectory, whereas the point-mass model uses the same standard values for most parameters (mass, power reduction, target speeds) for all aircraft. The linear regression method is efficient, although not as efficient as the Loess regression method. From an operational point of view, the proposed methods could be applied to the detection of potential conflicts between trajectories. Standard regression methods could be used to provide a relatively narrow probabilistic interval allowing us to detect conflicts with a great look-ahead time. In future works, we shall try to improve the Loess approach by introducing elements of the point-mass model in the predictors, and by testing other robust methods. Since the regression models had efficient results, we can use bigger dataset and random effect regression models to have a production level model working with different aircraft types, destinations and trajectory prediction phases. Another parallel plan could be to conduct a more thorough analysis of the available data, to obtain less noisy data. We could then learn the aircraft trajectory in a specific operation mode, thus giving a better chance to the point-mass model. It could also be interesting to learn some of the point-mass model parameters (mass, thrust law) from the observed data [17].

REFERENCES

[1] SESAR Consortium. Milestone Deliverable D3: The ATM Target Concept. Technical report, 2007.

[2] H. Swenson, R. Barhydt, and M. Landis. Next Generation Air Transportation System (NGATS) Air Traffic Management (ATM)-Airspace Project. Technical report, National Aeronautics and Space Administration, 2006.

[3] R. Alligier, D. Gianazza, and N. Durand. Ground-based estimation of the aircraft mass, adaptive vs. least squares method. In *ATM 2013, Chicago, USA, June*, 2013.

[4] X. Prats, V. Puig, J. Quevedo, and F. Nejjari. Multi-objective optimisation for aircraft departure trajectories minimising noise annoyance. *Transportation Research Part C*, 18(6):975–989, 2010.

[5] G. Chaloulos, E. CrÃijck, and J. Lygeros. A simulation based study of subliminal control for air traffic management. *Transportation Research Part C*, 18(6):963–974, 2010.

[6] N. Durand, J.M. Alliot, and J. Noailles. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the Symposium on Applied Computing, Philadelphia*. ACM, 1996.

[7] F. Drogoul, P. Averty, and R. Weber. Erasmus strategic deconfliction to benefit sesar. In *Proceedings of the 8th USA/Europe Air Traffic Management R&D Seminar*, June-July 2009.

[8] B. Musialek, C. F. Munafo, H. Ryan, and M. Paglione. Literature survey of trajectory predictor technology. Technical report DOT/FAA/TC-TN11/1, Federal Aviation Administration William J. Hughes Technical Center, Atlantic City, November 2010.

[9] C. Gong and D. McNally. A methodology for automated trajectory prediction analysis. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004.

[10] R. A. Vivona, M. M. Paglione, K. T. Cate, and G. Enea. Definition and demonstration of a methodology for validating aircraft trajectory predictors. In *AIAA Guidance, Navigation, and Control Conference*, 2010.

[11] J. Romanelli, C. Santiago, M. Paglione, and A. Schwartz. Climb trajectory prediction software validation for decision support tools and simulation models. *International Test and Evaluation Association*, 2009.

[12] Study of the acquisition of data from aircraft operators to aid trajectory prediction calculation. Technical report, EUROCONTROL Experimental Center, 1998.

[13] Adapt2. aircraft data aiming at predicting the trajectory. data analysis report. Technical report, EUROCONTROL Experimental Center, 2009.

[14] R. A. Coppenbarger. Climb trajectory prediction enhancement using airline flight-planning information. In *AIAA Guidance, Navigation, and Control Conference*, 1999.

[15] Christophe Bontemps. Prévision stochastique de trajectoires : procédures paramétriques et non-paramétriques. Master's thesis, Rapport de DEA IFP, 1997.

[16] I. Lymperopoulos, J. Lygeros, and A. Lecchini Visintini. Model Based Aircraft Trajectory Prediction during Takeoff. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Keystone, Colorado, August 2006.

[17] Richard Alligier, David Gianazza, and Nicolas Durand. Energy rate prediction using an equivalent thrust setting profile. In *5th International Conference on Research in Air Transportation (ICRAT 2012), May 22-25, 2012, University of California, Berkeley, USA*, 2012.

[18] Mohammad Ghasemi Hamed, Mathieu Serrurier, and Nicolas Durand. Possibilistic knn regression using tolerance intervals. In *IPMU 2012, Catania, Italy, July*, volume 299 of *Communications in Computer and Information Science*. Springer, 2012.

[19] Mohammad Ghasemi Hamed, Mathieu Serrurier, and Nicolas Durand. Simultaneous interval regression for k-nearest neighbor. In *Australasian Conference on Artificial Intelligence*, pages 602–613, 2012.

[20] W. S. Cleveland and S. J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *J. Amer. Statistical Assoc.*, 83(403):596–610, 1988.

[21] F. Imado T. Kinoshita. The application of an uav flight simulator - the development of a new point mass model for an aircraft. In *SICE-ICASE International Joint Conference Conference*, 2006.

[22] A. Nuic. User manual for base of aircarft data (bada) rev.3.7. Technical report, EUROCONTROL, 2009.

[23] Norman R. Draper and R. Craig van Nostrand. Ridge regression and james-stein estimation: Review and comments. *Technometrics*, 21(4):451–466, 1979.

[24] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Stat.Society. Series B (Methodological)*, 58(1):267–288, 1996.

[25] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Ann. Stat.*, 32(2):407–451, 2004.

[26] Engineering statistics handbook, 2011.

[27] M. I. Jordan and C. Bishop. *Neural Networks*. CRC Press, 1997.

[28] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1996. ISBN: 0-198-53864-2.

[29] B. D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996. ISBN: 0-521-46086-7.

[30] Y. Le Fablec. *Prévision de trajectoires d'avions par réseaux de neurones*. PhD thesis, Institut National Polytechnique de Toulouse, 1999.

[31] R.L. Eubank. *Nonparametric Regression and Spline Smoothing, Second Edition*. Statistics: A Series of Textbooks and Monogrphs. Marcel Dekker, 1999.

[32] T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models: T. J. Hastie and R.J. Tibshirani*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall, 1990.

[33] W. Härdle. *Applied nonparametric regression*. Econometric Society Monographs (No. 19). Cambridge University Press, 1990.

[34] G. Wahba. *Spline models for observational data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.

[35] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66*. Monographs on Statistics and Applied Probability, 66. Chapman & Hall, 1996.

[36] C. J. Stone. Consistent nonparametric regression. *Ann. Stat.*, 5(4):595–620, 1977.

[37] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Statistical Assoc.*, 74(368):829–836, 1979.

[38] J. Fan. Design-adaptive nonparametric regression. *J. Amer. Statistical Assoc.*, 87(420):998–1004, 1992.

[39] J. Fan. Local linear regression smoothers and their minimax efficiencies. *Ann. Stat.*, 21(1):196–216, 1993.

[40] Jianqing Fan and Irene Gijbels. Variable bandwidth and local linear regression smoothers. *The Annals of Statistics*, 20(4):pp. 2008–2036, 1992.

[41] D. Ruppert and M. P. Wand. Multivariate locally weighted least squares regression. *The Annals of Statistics*, 22(3):pp. 1346–1370, 1994.

[42] Q. Li and J.S. Racine. *Nonparametric econometrics: theory and practice*. Princeton University Press, 2007.

[43] Christopher G. Atkeson, Andreww. Moore Y, and Stefan Schaal Z. Locally weighted learning. *Artificial Intelligence Review*, pages 11–73, 1997.

[44] C. R. Rao and H. Toutenburg. *Linear Models: Least Squares and Alternatives (Springer Series in Statistics)*. Springer, July 1999.

BIOGRAPHIES

Mohammad Ghasemi Hamed currently a PhD student at the MAIAA lab of ENAC. M. Ghasemi Hamed is currently studying regression methods (probabilistic and possibilistic) for the ground based aircraft trajectory prediction problem. He received his engineers degree from ENSEEIHT (2010). He has also a graduate degree in M.Sc. (2010) in Artificial Intelligence from the University of Paul Sabatier in Toulouse and an undergraduate degree in computer engineering from the University of Shahid Beheshti in Tehran, Iran.

David Gianazza received his two engineer degrees (1986, 1996) from the french university of civil aviation (ENAC) and his M.Sc. (1996) and Ph.D. (2004) in Computer Science from the "Institut National Polytechnique de Toulouse" (INPT). He has held various positions in the french civil aviation administration, successively as an engineer in ATC operations, technical manager, and researcher. He is currently associate professor at the ENAC, Toulouse.

Mathieu Serrurier Ph.D. (2005) in Computer Science from the University of Paul Sabatier in Toulouse, He is currently associate professor at this university. His researches are related to uncertainty modeling and their application in machine learning. He had several works on the possibilistic regression and classification.

Nicolas Durand graduated from the Ecole polytechnique de Paris in 1990 and the Ecole Nationale de l'Aviation Civile (ENAC) in 1992. He has been a design engineer at the Centre d'Etudes de la Navigation Aérienne (then DSNA/DTI R&D) since 1992, holds a Ph.D. in Computer Science (1996) and got his HDR (french equivalent of tenure) in 2004. He is currenlty professor at the ENAC/MAIAA lab.