Flight Data Monitoring Application to Automatic Approaches and Landings

Yannick de Matos Lélis Duarte
yannick.duarte@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2017

Abstract

The Flight Data Monitoring department is an essential part of the commercial aviation management, that is why continuous improvement has been the guiding principle. The main goal of this project was to be a contributing factor to the safety of an airline operations, namely in the approach and landing phase of the flights, in the case when, certainly due to low visibility conditions, those approaches and landings were performed using the automatic flight mode (autopilot). An algorithm to classify automatic approaches and landings was created, and their data were stored in a local database along with the meteorological information, so that the regulators requirements would be fulfilled. Two supervised learning models, SVM and Naive Bayes, were created and trained, and the performance of their classification of the approaches and landings was evaluated. The project main contribution, lies on the development of a Shiny® application, that allows the stored data to be presented in a clear and informative way, along with the possibility to generate plots, access METAR information and crew flight reports.

Keywords: Flight Data Monitoring, Automatic Approaches and Landings, Classification Algorithm, Analysis Tool

1. Introduction

To help mitigating the operational risks, operator’s own Flight Data Monitoring (FDM) and Flight Operations Quality Assurance (FOQA) programs are the best potential sources of operational data, and they make use of flight data to identify trends and potential hazards in flight operations. Unstable approaches and hard landings represent a significant contributory factor in commercial aviation accidents that occur during the approach and landing flight phase, which make these two flight phases the most relevant ones to be monitored by the airlines safety department. This project takes the responsibility of building an algorithm to ensure that, all the automatic operations performed by the entire fleet is specially monitored, their performance is classified, and the data is duly stored to be used in future audits or routine checks [9]. The automatic approach and/or landing may have been made due to low visibility conditions or simply for training purposes. The meteorological information (METAR) is also collected and associated to each flight, and the two situations are then properly distinguished in further analyses.

Within the work developed, the FDM program with respect to stabilized approaches will normally assist in, (1) Monitoring of the flight parameters used to define stabilized approaches, (2) Understanding the factors contributing to unstabilized approaches, (3) Identifying correlations between unstabilized approaches and specific airports/runways, individual pilots, specific fleets, etc. The Flight Safety Foundation concluded that a stable approach is 60 times safer than an unstable one [14]. The Accident Classification Task Force (ACTF) allocates the factor “Unstable Approach” to an accident when “it has knowledge about vertical, lateral or speed deviations in the portion of the flight close to landing” [12].

Figure 1: Percentage of accidents with unstable approaches as factor [12]

Which makes the flight data monitoring of the
approach and landing phase an essential factor to guarantee flight safety.

2. Background

2.1. Flight Data Monitoring

Mandatory since January 2005, for aircraft over 27 tonnes [4], the FDM programs encompass the use of data generated during the flight for the purposes of Flight Operations Quality Assurance (FOQA) and Maintenance Operations Quality Assurance (MOQA), that allow their operators to identify areas of operational risk and quantify current safety margins, so that they can put in place appropriate risk mitigation to provide remedial action once an unacceptable risk has been identified and confirm the effectiveness of any remedial action by continued monitoring, making deviations from flight manual limits and standard operating procedures to be detectable, preventing hazardous events from happening, routinely monitor for engine trends that can indicate the need for engine maintenance, thus improving the airline’s overall safety key indicators, increasing maintenance effectiveness and reducing operational costs. FDM programs have an intrinsically non-punitive and just safety culture, where the incident data is used to reinforce training programs, raising awareness amongst the pilots group as a whole.

2.2. Flight data recording

The original purpose of recording data from flights was for post-incident investigation [6], however, airlines nowadays use it in a pro-active fashion. So in addition to the Flight Data Recorder (FDR), an aircraft flying today can have two non-crash-survival recorders, the Quick Access Recorder (QAR) and the Digital ACMS Recorder (DAR). The QAR records exactly the same parameters as the FDR, but it is, as the name implies, more accessible for a systematic collection of data, because it is located in a more suitable location in the avionics bay which is in the forward part of the aircraft.

The DAR is physically identical to the QAR and it is located also in the avionics bay. It is a more customized unit, where the set of parameters to be recorded is established by what the airline wants and can later on be modified at its need.

2.3. Analysis Ground System

AGS® is a SAFRAN Group - SAGEM Défense et Sécurité modular package system running on PC computer for flight data analysis, that concerns flight operations and engine maintenance departments. Due to its capability for input frame customization, AGS is capable of accepting data from all commercially available FDR, QAR and DAR sources on the market and automatically analyze all flight data available, making it the preferred choice of airlines. AGS is used to convert the binary raw data from ARINC 717 [3] protocol into aircraft parameters in engineering units, for further display, programming and analysis [17].

3. Automatic Flight Analysis

Automatic flight mode, or autopilot as it is usually designated, is the automatic control and guidance of the aircraft during some flight phases, where some tasks such as maintaining an altitude, climbing or descending to an assigned altitude, performing a precision or non-precision approach and landing, are automated [15].

Flight data from each flight, collected by the flight recorders, was processed in AGS, where the automatic approaches and landings were separated from the manual ones and exported to be analyzed in Rstudio® using R language. After being analyzed and classified into stabilized or unstabilized using Rstudio®, the automatic approaches and autolands are stored in a local database. Already with the flights stored, a Shiny Application is used to display and analyze the flights, for further analysis by the flight data analysts or for audits carried out by the regulators.

2.2. Flight data recording

The original purpose of recording data from flights was for post-incident investigation [6], however, airlines nowadays use it in a pro-active fashion. So in addition to the Flight Data Recorder (FDR), an aircraft flying today can have two non-crash-survival recorders, the Quick Access Recorder (QAR) and the Digital ACMS Recorder (DAR). The QAR records exactly the same parameters as the FDR, but it is, as the name implies, more accessible for a systematic collection of data, because it is located in a more suitable location in
pilot, thus giving an approach path that is perfectly aligned with the runway centerline thus allowing the aircraft to perform automatic approach and landing. It is equipped with three different types of transmitters on ground: (1) the localizer, (2) the glide slope and (3) two or three marker beacons.

(a) Localizer radiation pattern  (b) Glide slope radiation pattern

Figure 4: ILS glide slope antenna and radiation pattern

Associated to the ILS navigation there are 3 different categories, that have decision heights and runway visual range minima, that define in which category the ILS navigation is being made. The autoland flight mode procedure is only mandatory when the crew is presented with the CAT III [1] condition.

Figure 5: ILS approach and landing categories

3.2. Automatic Approach and Autoland Classification

The best way to understand unstable approaches and landings, is by understanding stable ones. An approach is considered to be stable if it meets the stabilized approach criteria defined by the operator in its Standard Operating Procedures (SOPs), and in the case where the situation cannot be reversed, a go-around must be initiated by the pilots, because otherwise it may result in the aircraft touching down too fast, too hard, outside the touchdown zone (long or short), off the runway centerline, in the incorrect attitude or incorrectly configured for landing.

The main challenge is to effectively define a common set of parameters that constitute a stabilized approach, which should be maintained until touchdown. This will help to ensure that all stakeholders will be working to achieve the same outcome, but due to the wide variety of aircraft types, the different environmental constraints, the airport conditions, and the airlines operational needs, this common criteria is not always easy to settle. Nevertheless, in general and conceptual terms these criteria are essentially the same. From this common criteria concept, came out that the aircraft must have, at the stabilization height, the right configuration, attitude, airspeed, power/thrust setting, and be at the right position over the runway, which means no excessive ILS glide slope or localizer deviations, in the instrument landing procedure [2], [10], and this is achieved by monitoring a set of key parameters, defined by the airline in the Flight Crew Operating Manuals [18] and [19], in compliance with the common criteria.

1. The aircraft is on the correct lateral and vertical flight path.
2. Pitch, Yaw and Roll angle values do not vary so much in order to maintain the desired flight path.
3. The aircraft is in the desired landing configuration.
4. The thrust is stabilized, usually above idle, to maintain the target approach speed along the desired final approach path.
5. The landing checklist has been accomplished as well as any required specific briefing.

3.3. Meteorological Conditions Information

To be able to distinguish the situation when the automatic flight mode was used due to adverse meteorological conditions from when it was used just for training or routine cases, and also to verify how adverse were the weather conditions when the autoland or the automatic approach was performed, Aviation Routine Weather Reports (METARs) were collected from the National Oceanic and Atmospheric Administration (NOAA) Aviation Weather Center (AWC) website.

For a better understanding on how the METAR information is decoded, the reader is advised to consult [5]. It was developed an algorithm which uses the NOAA website and extracts all the METAR information related to all TAP destination routes, and this way all the information would be stored in TAP’s own database, until it is no longer necessary. This same algorithm decodes the eXtensible Markup Language (XML) information, and delivers the METAR raw text along with the appropriate description.

We have then, as final outcome, the automatic flight stored in the local database with the respective meteorological information that was registered 30 minutes or 1 hour before the landing.
4. Classification Algorithm

After properly separating the automatic approaches from the manual ones, a set of recorded and computed parameters relative to those flights, were chosen to be exported as csv files from AGS® and further to be processed in Rstudio® using R language, where a classification is made to distinguish the automatic approaches with manual landing from the ones with automatic landing. Once more, the autopilot indicator is the key parameter to determine that: (1) if at touchdown any of the autopilot indicator parameter is still engaged, then we have an autoland, (2) otherwise we only have an automatic approach followed by a manual landing, and in this case, only the automatic segment of the approach is analyzed.

![Figure 6: Daily task to be implemented to extract, classify and store the automatic flights](image)

The automatic procedure performance classification can be represented as a disjunction of the seven key parameters selected according to the criteria, where the “Performance” will be either “1” or “0” depending on whether any of the parameters took the value “1” or not.

![Figure 7: Performance Classification Logic](image)

Beyond the key flight parameters, also wind limitations present in [18] and [19] were monitored, and any deviations from the specified values were duly noted, which can help to relate their deviations with the deviations of the other parameters.

4.1. Gaussian Smoothing

Dealing with numerical data that comes from sensors measurements, brings the problem of bad measurements due to the presence of noise, and outliers might arise due to this. In order to make the data smoother and less noisy, filtering techniques are normally required to remove additional noise which can degrade the quality of the desired portion.

The Gaussian window is a low pass digital filter technique that uses the Gaussian function (known as the bell curve) [11], to remove noise and also details of the observations. In one dimension, the Gaussian function, or the zero mean normal distribution, is given by:

\[ g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \]  

(1)

Where \( \sigma \) is the standard deviation of the distribution. And for a variance of \( \sigma^2 = 1 \) we have:

![Figure 8: Gaussian or bell curve function with zero mean](image)

As we can see from Figure 8, the Gaussian function extends to infinity and so it must be truncated at the end of the window, and since the log of a Gaussian produces a parabolic, it can be used for exact quadratic interpolation in frequency, and the coefficients can be computed from:

\[ w(n) = \exp\left[-\frac{1}{2} \left(\alpha \frac{n}{(N-1)/2}\right)^2\right] \]  

(2)

where, \(-\frac{N}{2} \leq n \leq \frac{N}{2}, N - 1 \) being the window length and \( \alpha \) the window width, which is inversely proportional to the standard deviation \( \sigma \), so using a greater value for \( \sigma \) produces a narrower window.

Knowing that,

\[ \sigma = \frac{N - 1}{2\alpha} \]  

(3)

Then, we can write a simpler equation to compute the coefficients:

\[ w(n) = \exp\left(-\frac{n^2}{2\sigma^2}\right) \]  

(4)
In order to filter the sampled data, we need to compute the convolution between the data and the Gaussian function.

\[
y(t) = x(t) * g(t) = \int_{-\infty}^{\infty} x(\lambda) g(t - \lambda) d\lambda
\]

\[
y(n) = \sum_{\lambda=-\infty}^{\infty} x(\lambda) g(n - \lambda)
\]

Equation 5 represents the convolution of two continuous functions with the limit to infinity, which for this case can’t be applied, and so we need to compute the discrete convolution, in 6, between the sampled parameter and the sampled Gaussian kernel that is produced by sampling points from the continuous Gaussian.

\[
y(n) = \sum_{\lambda=-\infty}^{M} x(n - \lambda) g(\lambda)
\]

where \(M\) is the gaussian window length.

It is illustrated in Figure 9, how can the data from a recorded parameter be improved by the Gaussian smoothing technique, using different values for \(\sigma\), the tuning parameter, we can find a balance between, smoothing enough to “clean up” the noise but not so much as to remove important information in the data.

![Figure 9: Smoothed observations of the pitch angle](image)

(a) \(\sigma = 2.5\)  
(b) \(\sigma = 5\)  
(c) \(\sigma = 7.5\)

5. Support Vector Machine

Support Vector Machine is a supervised learning method, that is related to binary classification problems, where it separates the data in 2 groups [8]. For being a supervised method, it will need a finite sample training set, let’s say \(S = (x_1, y_1), \ldots, (x_n, y_n)\) with \(x_i \in \mathbb{R}, y_i \in \{0, 1\}\), where in this project \(x\) will be the flight parameters (\(n\) features), that will define their classification \(y\), 1 for Unstable, and 0 for Stable.

Supposing that there is a data set drawn from an underlying probability distribution \(P\), one needs to estimate a subset \(\hat{S}\) of the input such that the probability that a test point from \(P\) lies outside \(\hat{S}\) is bounded by some prior specified value. So the solution for this problem is to obtain by estimation, a function \(f\) which is positive in \(\hat{S}\) and negative in the complement of \(\hat{S}\). After building the models, the goal will be to have an estimator function \(f\) that correctly predicts the label \(y_i\) for a given input vector \(x_i\).

\[
f : X \in \mathbb{R}^m \rightarrow Y \in \mathbb{R}^k
\]

There are two different situations that may happen when dealing with binary classification of the data: 1 - A simple situation, where the data can be separated by an hyperplane without any training errors, thus considered to be “linearly separable”, or 2 - A less simple one, where data is not linearly separable and cannot be separated by any hyperplane, the “non-separable data”.

Because we are dealing with numerical data, from measurement units, we are inevitably surrounded by noisy samples and by the presence of outliers, which makes the data not linearly separable, thus not allowing any hyperplane to correctly classify all the observations. The data is required to satisfy \(y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i\), where \(w\) is a normal vector to the hyperplane, and also there is the need to use a cost parameter \(C\), which penalizes training errors. We will then have a new expression for the objective function defined as:

\[
\text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_i \xi_i
\]

Subject to \(y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0\)

where the ordered pair \((p, k)\) is either \((1,1)\) or \((2,2)\). The Lagrangian is then given by:

\[
L(w, \xi, \alpha, \beta) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i [y_i (\langle w, x_i \rangle + b) - 1 + \xi_i] - \sum_{i=1}^{n} \beta_i \xi_i
\]

with \(\alpha\) and \(\beta \geq 0\)

\[
\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i
\]
and the second derivative about \( w \) is still unitary, which makes the optimization problem to find minimum value for \( w \):

\[
\frac{\partial^2 L}{\partial w^2} = 1
\]

(11)

\[
\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0
\]

(12)

\[
\frac{\partial L}{\partial \xi} = C - \alpha_i - \beta_i = 0 \Rightarrow \alpha_i = C - \beta_i
\]

(13)

Using these relationships, yields:

\[
W(\alpha, \beta) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle
\]

\[
+ C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} \beta_i \xi_i
\]

\[
= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle
\]

(14)

with:

\[
0 \leq \alpha_i \geq C
\]

(15)

With all the lagrange multipliers varying between 0 and \( C \), the set of possible \( \alpha \) forms a hypercube. And the Karush-Kuhn-Tucker (KKT) [13] complementarity conditions for this problem are:

\[
\xi_i (\alpha_i - C) = 0
\]

(16)

\[
\alpha_i [y_i (\langle w, w_i \rangle + b) - 1 + \xi_i] = 0
\]

(17)

To be able to apply the SVM models into the nonlinear flight data acquired from the flight recorders, one needs to map them into a complete inner product space, a Hilbert space (the same as Euclidean space), using a nonlinear map.

So if, \( X = \{x_1, ..., x_l\} \), and the desired feature space is \( F \), then \( \phi : X \rightarrow F \) will be the mapping operation, using the nonlinear map \( \phi \).

With the transformed data, we can then apply the SVMs, which will have \( S = \{\phi(x_1), ..., \phi(x_l)\} \) as training set, resulting in the decision function:

\[
f(x) = \text{sign} (\langle w, \phi(x) \rangle + b)
\]

(18)

The Lagrange multipliers will continue to identify points that are on the margin hyperplanes, and \( w \in F \) will represent the linear combination of the training set \( S \):

\[
w = \sum_{i=1}^{n} \alpha_i y_i \phi(x_i)
\]

(19)

Rewriting the decision function for the nonlinear case using the linear combination 19, yields:

\[
f(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + b \right)
\]

(20)

which, as expected, is nonlinear in \( \mathbb{R}^n \).

Although well defined, the nonlinear SVM still has some distress, when choosing the best \( \phi \), and even after that, the computation of the inner product in the featured space. That’s where the importance of using Kernel functions arises, and their use obliterates the need of doing either one of the above mentioned steps.

A Kernel is a function \( K \), such that \( \forall x, z \in X, K(x, z) = \langle \phi(x), \phi(z) \rangle \) where \( \phi : X \rightarrow F \).

Then, the inner product in the nonlinear decision function can be replaced with \( K(x_i, x) \), which updates the classification (decision) function to:

\[
f(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \right)
\]

(21)

and being the support vectors the only Lagrange multipliers \( \neq 0 \), then:

\[
f(x) = \text{sign} \left( \sum_{SV} \alpha_i y_i K(x_i, x) + b \right)
\]

(22)

There are many options when it comes to choosing the type of Kernel function to use, but the most common ones are:

- Linear \( \Rightarrow K(x_i, x_j) = x_i^T x_j \)
- Polynomial \( \Rightarrow K(x_i, x_j) = (\gamma x_i^T x_j + \xi)^d, \gamma > 0 \)
- (RBF) Radial Basis Function \( \Rightarrow K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0 \)
- Sigmoid \( \Rightarrow K(x_i, x_j) = \tanh (\gamma x_i^T x_j + \xi) \)

During the experiments, three of these Kernels were used, and only the ones with better results were selected.
6. Naive Bayes Classification

The Naive Bayes classifier is a likelihood classifier, in light of Bayes’ hypothesis, it has a very high learning efficiency and can estimate all the probability after a total scan of the training data. Bayes’ hypothesis determines the probability of each class \( A \) given the values \( B_i \) of all attributes for an instance to be classified, and the conditional independence of the attributes for a given class are:

\[
P(A | B_1, ..., B_n) = \frac{P(A)}{\prod_i P(A | B_i)}
\]  
(23)

As we can see from equation 23, Naive Bayes classifier is based on the assumption that the principle of maximum posteriori hypothesis to identify the object that is most likely to be classified under the category [20], thus combining prior knowledge with observed data and then assigns a posterior probability to a class based on its prior probability and on its likelihood given the training data.

Hence, we get for a flight \( d \) and a class \( c \), the following classifier:

\[
c_{MAP} = \arg \max_{c \in C} P(c | d)
\]

\[
= \arg \max_{c \in C} \frac{P(d | c)P(c)}{P(d)}
\]

\[
= \arg \max_{c \in C} \left[ \prod_{i} P(x_i | c)P(c) \right]
\]  
(24)

where \( x_1, ..., x_n \) represent the flight features (parameters).

Naive Bayes classifiers are not so good when dealing with noisy features and binary classification, because it can only learn linear discriminant functions [16], which makes it always suboptimal for non-linearly separable data.

7. Results

7.1. General Results

Because the study is based on low visibility operations, the set of flights chosen to be analyzed, were automatic flights made from the 1st of December until the 1st of March, thus constituting a set of 19090 flights performed by the whole TAP fleet, within all routes and it was built an output table of the 19090 automatic flights divided by the three categories according to how the approach and landing were made.

Where type 1 is all the flights, 2-manual approach and landing, 3-automatic approach with manual landing, and type 4-autoland.

From Table 1, we can see that the situations where it was necessary the use of automatic approaches and the use of autolands, have been not so many, and the manual procedures were the prevailing ones. After this, it was then essential to signalize the amount of unstabilized cases for both cases, already with the smoothed data.

<table>
<thead>
<tr>
<th>Perf.</th>
<th>Total</th>
<th>Type</th>
<th>A319</th>
<th>A320</th>
<th>A321</th>
<th>A330</th>
<th>A340</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stab. (74.31%)</td>
<td>185</td>
<td>190</td>
<td>30</td>
<td>43</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unstab. (25.69%)</td>
<td>74</td>
<td>51</td>
<td>6</td>
<td>10</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Auto approaches with manual landing performance

Analyzing Tables 2 and 3, we can see that, curiously, the automatic approaches that didn’t result in autolands were the ones with the greatest number of unstabilized performance. That could be explained by the fact that for some reason, the automatic system was not stabilizing the approach, which made the pilots to decide upon taking the control of the aircraft and perform a manual final approach and landing.

For a better understanding of which were the factors that contributed the most for unstabilized automatic approaches and landings, at each flight were added extra columns of data to indicate which was/were the parameter(s) that contributed for it to be unstable.

<table>
<thead>
<tr>
<th>Type</th>
<th>Indicated Airspeed</th>
<th>Rate of Descent</th>
<th>Pitch Angle</th>
<th>Roll Angle</th>
<th>Localizer Deviation</th>
<th>Glide Slope Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>148</td>
<td>11</td>
<td>6</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 4: Number of exceedance cases for each parameter

7.2. Database

An automatic flights database was built, tables containing relevant data, one table for the general flight information, with one row per flight, thus containing all the automatic flights, five other tables containing the key parameters of each flight, and a final table for the METAR information related to the flights in question.

In short, we have 3 blocks of data stored in the local database, one block of general data that works...
as identifiers for each flight, another block that has the flight parameters data for the 5 variants of each aircraft type and engine, and a third block that has the METAR information of each flight stored. In the example of Figure 11, we can see how the different tables are connected between them, and how they get the information they need from each other, to be used later on the Shiny app, developed to display the data and to allow interactions.

7.3. Visualization and Analysis Tool

A visualization and analysis tool was developed using R programming language to code all the structure, using the features available, adapting them to what is essential to the data monitoring and interaction process, to meet the safety requirements. The tool is divided in two “tabs” that have distinct utilities, the first one is a more statistic oriented sector that displays an overview of the total number (time interval selected by the user) of automatic flights and their performance using a table that contains the key information for the flight identification of all the flights contained in the date range selected with a binary indicator for each key parameter, it is “0” if the parameter wasn’t exceeded, and “1” otherwise, and also indicated if a “Go Around” was made, then a pie chart, with the percentage of the flights that were performed using manual approach and landing, of automatic approaches with manual landing, and also of autolands. Four histograms follow up to display the total number of stabilized and unstabilized automatic approaches with manual landing or autolands, depending on which mode is selected in the radio buttons, the number of stable and unstable flights, distributed by the entire fleet, by the aircraft registration code (Aircraft Tail Number), and by all runway destinations at which the aircraft performed such approaches and landings.

the user can click on the corresponding flight and it will be forwarded to the second tab, that is a more parameter oriented section, that displays the exceedance plots and allows the user to change what he/she wants to see in other plots. It starts with a little table showing the basic flight information, flight number, the date, aircraft type, tail, origin and destination runways, and a clickable icon, that leads the user to the official crew flight report, that reveals the crew perception about the autoland performance, and the meteorological information, that was gathered using the method explained during this thesis, the METAR information is displayed as raw text followed by the description of each part of the METAR.

When the user clicks the flight in the previous tab, an automatic plot that shows the exceedance(s) related to the flight during the approach and landing is generated, and the analyst can visualize the exceedance and draw some conclusions.

Having only the plot of the exceeded parameters may be insufficient to fully understand why the approach or the landing was unstable, so it was developed a feature that allows the analyst to plot any of the parameters associated to the flight, in time or in height above runway, thus providing the analyst with a more complete vision of all other parameters which may have caused the parameters exceedances
and make the procedure to be unstable, and detect particularities of some segments.

With this Shiny application, built from the ground to what was presented above, the analysts in the safety department can now have a complete table of the automatic approaches with manual landing and of the autolands performed within a date range at their choice.

In addition to the statistical information, that the tool provides in the first tab, it also provides the user with a more technical functionality, as the METAR information or the ability to generate plots with the information on how much were the exceedances registered, which makes the application to be considered a very versatile tool.

### 7.4. Models Classification Results

Using R language and the LIBSVM software [7] package “e1071”, the models were trained using 795 flights (432 Autolands and 363 Auto Approach with Manual Landing) from 01-12-2016 to 31-01-2017, already classified as Stabilized or Unstabilized. The models created, were based only on the A319/A320/A321 family, and were divided into 4 models: (1) Autoland SVM model, (2) Autoland Naive Bayes model, (3) Auto approach with manual landing SVM model and (4) Auto approach with manual landing Naive Bayes model, and the automatic classification can be summarized in two steps: (1) Building the model with a labeled training set of flights and (2) Obtaining label predictions using a test set of flights.

With the models already built using a set of 795 flights (432 autolands and 363 auto approaches with manual landing) from 2016-12-01 to 2017-01-31, it was then selected 127 flights (34 autolands and 93 auto approaches with manual landing) from 01-02-2017 to 01-03-2017, and these flights were automatically labeled by the 4 models. To verify the accuracy of the models classification, the flights were also classified by the parameter inspection mode to evaluate the models accuracy for both autoland and automatic approach with manual landing.

#### Table 5: Autoland models performance

<table>
<thead>
<tr>
<th></th>
<th>NB Findings</th>
<th>Correct</th>
<th>SVM Findings</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stab</td>
<td>27</td>
<td>88.88 %</td>
<td>30</td>
<td>86.66 %</td>
</tr>
<tr>
<td>Unstab</td>
<td>7</td>
<td>57.14 %</td>
<td>4</td>
<td>100 %</td>
</tr>
</tbody>
</table>

Starting with the autolands, the NB found 7 unstable cases, but of those 7, only 4 were also classified the same way by the parameter inspection method, leaving 3 unclassified as unstable, and 3 misclassified. The SVM model found 4 unstable cases, and all those 4 were also found by the parameter inspection method, which means, only 3 were left unclassified as unstable.

For the automatic approaches with manual landing, the NB model found 31 unstable cases, but, only 20 were also found by the inspection method (total of 32), which means 9 were misclassified and 12 were left unclassified as unstable. The SVM found 23 unstable cases out of 32 found by the parameter inspection method, leaving 9 cases unclassified.

#### Table 6: Auto approach with manual landing models performance

<table>
<thead>
<tr>
<th></th>
<th>NB Findings</th>
<th>Correct</th>
<th>SVM Findings</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stab</td>
<td>62</td>
<td>80.65 %</td>
<td>70</td>
<td>87.14 %</td>
</tr>
<tr>
<td>Unstab</td>
<td>31</td>
<td>64.52 %</td>
<td>23</td>
<td>100 %</td>
</tr>
</tbody>
</table>

As one would expect, the models are not 100% accurate at labeling the flights, since they depend on the statistic factor and on the Euclidean distance of the data, for the Naive Bayes and the SVM case, respectively. Even in the cases where all the unstable findings were correct, there were always unidentified unstable flights.

The SVM models were the ones that obtained better results, as we can see in Tables 5 and 6. This can be explained by the fact that when it comes to noisy or non linear data, the Naive Bayes classifier does not behave in the best way, making it
to misclassify more flights than the SVM models, and the autoland models were the ones with better accuracy, probably due to the fact that its training set had a higher number of flights.

8. Conclusions
Throughout this project, it became clear the importance of having an FDM program consistently implemented in the operations of an airline, since it plays an important role in ensuring that the operations are performed under the best safety policies and conditions, and that the on-ground post flight analysis of the recorded flight data turned out to be the best way to mitigate risks and control hazardous trends within the most critical flight phases, like the approach and landing.

With the development of the tool, we were able to observe in a more global way, the statistics of the automatic flights, and realize the proportion of the unstabilized cases and the reason why they happened. It was also possible to perceive which airports/runways, fleets and the combination of both, contribute the most for the unstabilized cases, and overall, it is fair to conclude, that the automatic procedures, approaches and landings, have a solid success rate, in compliance with the stabilization criteria commonly established, thus having a general idea of where and with which aircraft, it is more likely to have a hazardous event.

The major achievement within the elaboration of this thesis, was the final version of the analysis tool, that can prove to be very useful within the FDM program of an airline, and continues to allow that the airline complies with the requirements imposed by the National Aviation Authority, of monitoring the number of automatic approaches and autolandings and their performance, with a better, improved and more complete tool, which also provides the meteorological information associated to the flight, that can help to better understand the story behind any unusual system behavior.

References