UAV Flight Dynamics: Design and Development of a Framework for Flight Data Processing and Analysis

Sean Bazzocchi
sean.bazzocchi@gmail.com

Instituto Superior Técnico, Universitade de Lisboa, Portugal

November 2018

Abstract

The Center for Aerospace Research, working on the leading edge of the next generation of unmanned aircraft, carries out an extensive activity of flight testing with the objective of studying new configurations and systems to fulfil missions in the most efficient way.
Since the acquisition of several projects by collaborators such as Boeing, Bombardier and the Canadian Department of Defence, the number of flight tests to be performed has increased substantially, requiring a radical change in the data handling method used to analyse the numerous log files generated. The project developed in this thesis proposes to solve this issue with the creation of a customized flight data post processor software, capable of managing all the data produced from the flight tests.
This software not only provides a unique tool to handle and process all the flight data, but also integrates a variety of features in order to extend the analysis capability. The functions implemented range from simple data extrapolation tools, to complex frequency analysis algorithms used to study the dynamic response of the aircraft. Moreover the software offers a sequence of functions to perform a complete tuning process of the autopilot with the objective of increasing the control efficiency, by removing the oscillatory response caused by erroneous parameters.
To complete the project a study of a new type of flight mission plan is conducted in order to improve the dynamic analysis of the aircraft performed by the advanced tools in the software.
Keywords: autopilot tuning, flight data analysis, test post-processor, Piccolo controller, aircraft transfer functions.

1. Introduction

Any aircraft can be observed as a system in terms of its input (i.e. controls commanded) and output parameters (i.e. sensors), even with out other information of it's internal working, as if it was a black box model[1] [1]. The advantages of assuming this new prospective, that removes the internal controller, pilot and/or autopilot, from the loop created during the actual flight process, emerges when no feed back of a decision made is possible or when the intention is to study the response of the aircraft in a more detailed way.

The lack of a pilot on the autonomous UAV's developed requires, for research flight testing purposes, the instalment of several sensors in order to keep track of the sensitive flight parameter values. The information from the sensors is generally collected by a flight data recorder (FDR) in a single log file, but due to the fact that the advance sensors used cannot always be connected to a communal hardware and/or the language of the data transmitted is different, a single unit is not capable of accomplishing the task, so different types of recording devices need to be installed. Each unit stores the data in different methods, with different formats and sampling rates, this causes the information to be spread among non compatible files.
Although multiple programs are provided by the different manufacturers to list and plot the log files produced by their sensors, these are limited by the fact that they could only process the parameters contained in the relative log file. Without the possibility to combine the data from other log files the potentiality of the post processing is constrained to a low level of complexity.
These conditions particularly aggravate the work needed to tune the automatic control system. This task requires the entire data from all the different devices in order to study the dynamic response of the aircraft in every flight phase. Moreover the fact that the UAV's are developed in a research struc-

[1] The black box model, not to be confused with the informal synonym of flight data recorder, describes a system defined only by the inputs and outputs (I/O) without any knowledge of the internal architecture or process

ture imply that the a/c configuration is subject to different mutations in a short period of time, requiring a diverse tuning of the autopilot for every change [2].
Coupled with the advance processing and the autopilot tuning tasks, the consignment of the flight data to partners or customers is also hindered by the fragmented state of the information.
From the state of data handling described it was evident that a new process needed to be developed in order to overcome these issues.

2. Software framework

The general structure of the software developed reflects the organized groups of requirements defined during the initial design of the software. The requirements were sorted into 5 macrogroups, each one of these represent a generation of the software and is indicated by the the first digit of the release number also known as major. Every major version of the software is built on a series of minor versions that implement and test singular functions. The ensemble then operates to meet the general requirement.
The subdivision of the functions is generated in the software with the a Tabbed Document Interface[2] (TDI) organizing the different tools into 6 different panels[3]. The tabs, as shown in Figure 1, are placed in the top section of the window in the sequence of the workflow that must be developed to analyse a full flight.

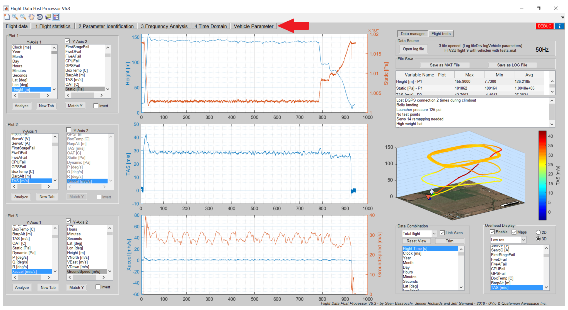


Figure 1: General overview of the graphical interface with TDI.

The tabbed structure, due to the lack of a dedicated library in Matlab Graphical User Interface Development Environment (GUIDE), was manually developed with the combination of toggle-buttons and panel visibility controls, deactivating features not related to the toggled tab.
When the software is launched, the first tab *Flight data* is active. This is the general panel in which the user can open and load flight logs, perform initial processing, list the variables, plot the data and observe the flight path.
The next tabs are enumerated from 1 to 4 creating a sequenced workflow for the advance post processing task. In step 1, the panel *Flight statistics*, identifies all the general values of the flight to be stored as statistic data for design factors. *Parameter identification* (step 2) analyses the possible test points of the flight to calculate some of the control coefficient of the autopilot. Step 3 of the post processor is the *Frequency analysis*, here the transfer functions (TF) are created from the current dynamic coefficients and studied in the frequency domain. The TF generated are then compared in the *Time domain* (step 4) with the real time history response of the aircraft that allows to determine the optimal tuning of the autopilot.
Vehicle parameter is the last tab and, as the first one, has no number since it is used within the post processing flow to check the overall status of the parameters. However it is placed at the end, since it closes the post processing work with the exporting feature of the vehicle parameters configuration file, to be then uploaded in the autopilot.

2.1. Flight data

The first tab interface is capable of loading, merging and processing log files with the purpose of visualizing, extrapolating and analysing the flight data.
The loading feature is able to scan log files from the flight controller, ground station telemetry, and external sensors. It also performs an error detection scan to correct common logging errors or to warn the user about the presence of unusual faults.
The import function, unlike the software provided by Cloud Cap, was written with a formatted scan feature which increases the speed by 128 times compared to the provided one. The data from the logs is combined by a resampling function that up-samples the low-rate logs, time shifted comparing the T/O acceleration trend and processed to obtain extra parameters from the combination of the logged variables [3, 4]. The processed data is then plotted on the GUI with the possibility to set variables, custom view options, zoom, pan and trim the data.
All the controls for navigating in the flight records are customized functions that trigger other features. For example zooming or panning sets the time frame of the data visualized, updating the flight path display and highlighting the current flight section visualized. The mentioned flight path display is a useful tool that helps the user to identify the section of the flight corresponding to the data framed

[2]TDI, or more generally called Tab, is a graphical control element that allows multiple panels to be contained in a single window.
[3]Although the macrogroups stated were 5, the software is structured with 6 tabs because the increment cycle named *Vehicle parameter integration and parameter identification* is a combination of two macrocroups that needed to be released jointly due to a strong dependency.

in the time history plots.
A stats table and a data cursor function are present to perform basic data extrapolation, displaying maximum, minimum, average and instantaneous values across all selected variables. The flight test point identification feature allows the user to search and define a specific test performed during the flight (e.g. stall, doublets and chirp commands) that can then be processed in the advanced analysis sections of the software.
All the processed data and annexed information can be saved in a single binary file, generating a unique file for each flight. The advantages of this new saving system are numerous, from the improvement in the organization of the flight records to the capability of storing all the processing action performed and additional parameters.

2.2. Flight statistics

The need for general flight data, expressed firstly by the design team for development parameters and secondly by the customers to check the flight tests progression, led to the development of an automated tool to standardize the extrapolation process of a predefined set of data to be shared outside the software environment. This new feature not only speeds up the process for the flight analysis specialist but also improves the communication from the flight testing department to the other sections providing an accurate statistic trend of the most demanded parameters during flight.

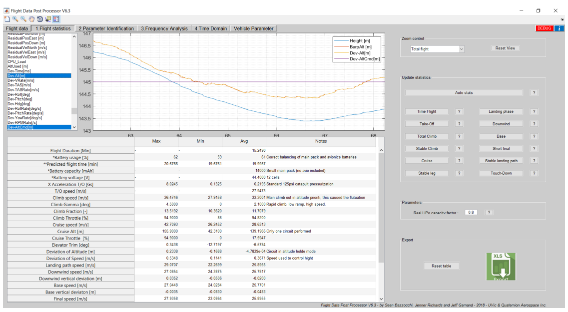


Figure 2: Flight statistic tab interface.

The flight statistics interface presenter in Figure 2, contains all the functions needed to identify, calculate, display, modify and export the general parameters for the statistic trend.

2.3. Vehicle parameters

In order to develop the advanced processing tools used to analyse the dynamic response of the UAV and optimize the tuning of the autopilot, it was necessary to build an organized data structure in which all the parameters that identify the aircraft and controller characteristics were allocated.
It was during the development of the *Parameter identification* tool that not only the need of the structure created was discovered, but also the need of a complete interface system that allowed the interaction and visualization of the coefficients. For this reason the *Vehicle parameter* development cycle was performed in parallel, and deployed before the parameter identification release.
The toggle button of the vehicle parameter interface is located in the last position of the tab because it does not actively participate in the advance processing of the dynamic data, also the function *Export parameters* featured in the interface represents the final action in the advance processing workflow, this sets the conditions for the next flight test.

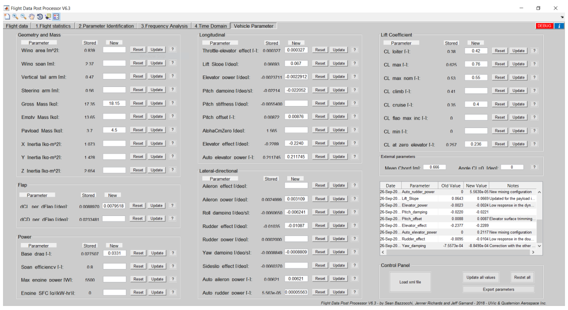


Figure 3: Vehicle parameter tab interface.

The interface, shown in Figure 3, displays the vehicle parameter structure, a complete set of coefficients which characterize the controller. The vehicle parameters, extrapolated from the XML configuration file of the a/p, are stored in a double structure which allows the user to define a second set of temporary values for each coefficient. This system gives the ability to perform a direct comparison analysis of the controller tuning. Also a history table keeps track of the modified parameters, granting the possibility to retrieve the initial values and helping the user to understand the previous adjustments with a side note.

2.4. Parameter identification

The *Parameter identification* tab represents the first step of the advanced post processing aimed at the optimization of the autopilot gains [5]. It features several tools for the analysis of predefined test points and the calculation of controls coefficient and aerodynamic parameters.
To perform the analysis, a set of predefined tests are required to be performed during the flight. These points are identified by the time frame and classified by the test-type in the flight-log thanks to the *test point identifier* tool present in the flight data tab. From each test, with the differential response calculation, it is possible to

analyse two states commanded and approximate the aircraft behaviour after a surface deflection or a state change. The new parameters calculated can then be temporarily saved in the vehicle structure and compared with with the autopilot in-flight configuration.

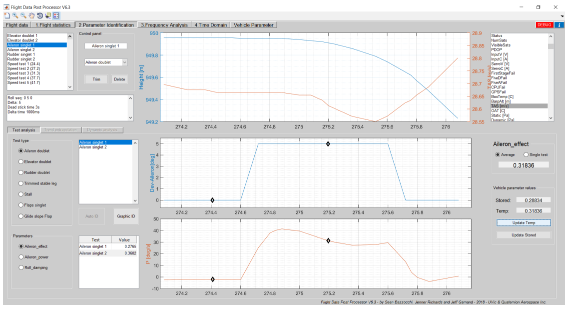


Figure 4: Parameter identification tab interface.

The interface, shown in Figure 4, is clearly divided in two sections. The upper one is dedicated to the management of the test points, while the inferior section collects the tools for the test analysis.

2.5. Frequency analysis

At this stage of the advanced post processing, the software should have guided and helped the user to identify some of the vehicle parameters in order to improve the tuning of the controller. Unfortunately not all the parameters needed can be identified from test points and, furthermore, even the ones calculated cannot be directly uploaded into the autopilot due to the fact that they could be subject to errors or dependency from other coefficients.
To solve this problem the *frequency* and *time domain* analysis tools were designed. They both employ an algorithm, that uses the vehicle parameters to generate a double set of transfer functions (TFs) that can then be imported and studied in both domains, to verify the improvements produced by the new identified parameters and optimize the remaining coefficients to grant dynamic properties and match the predicted response with the real tested one [6].

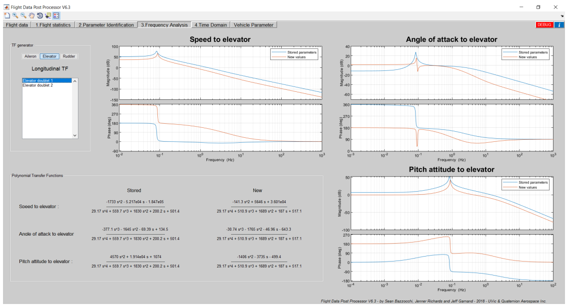


Figure 5: Frequency analysis tab interface.

The frequency analysis tab, shown in figure 5, represents the first step of the TF study [7]. In this interface the transfer functions are calculated, displayed and plotted in the frequency domain.
The interface is very simple, it only has a few controls that allow to define the type, and the test, to analyse. The selection of one item from the listbox triggers the *TransferFunctionGenerator* script which outputs the TFs to be displayed firstly as polynomial functions and secondly analysed with the bode plot tool.

2.6. Time history identification

Finally, the *Time domain* tab completes the overview of the software framework.
This last tool was generated with the purpose of studying the time domain response of the transfer functions in order to analyse the differences between the theoretical solutions and the real flight data. This is possible using software simulations where the initial flight parameters and the controls deflection are given as input to the TF which calculates the predicted dynamic response [8].
The tab interface is shown in Figure 6 where a doublet elevator test is analysed.

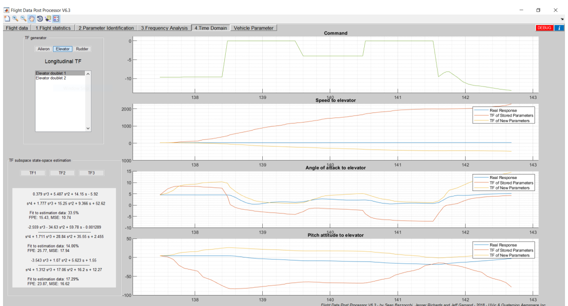


Figure 6: Time domain analysis tab interface.

The interface, like the frequency analysis tab, is very basic. It features the time history response plot with the commanded deflection in the main part of the window, while on the left side the control panel and the transfer function estimator are present. The selection of a test point in the

control panel on the left side of the plots triggers two functions of the tab, the first is responsible for the TF generation and simulation, the second plots the responses on the axes.
The time history comparison of the two sets of TFs with the real response of the test performed, is a powerful tool that can immediately show the accuracy of the controller to predict the dynamic response of the aircraft and help the tuning of the vehicle parameter in order to match the two curves. The time domain panel also features a transfer function state-space estimator, which generates the best matching TF containing a given number of poles and zeros, using the real response as discrete-time data [9].

In conclusion, when a new set of parameters have been proven to be more efficient in the response estimation of the UAV, the vehicle structure can be exported from the vehicle parameter tab in a copy of the original XML file and loaded directly in the Piccolo controller for the next flight test.
Despite the fact that these advanced analysis tools are very powerful for the tuning of the autopilot, they must be used consciously due to the fact that the transfer functions used by the simulation software are not exactly the same as those coded in the controller[4].

3. Flight test planning

Before the deployment of each software version a testing process is performed to validate the implemented features. Since the software is a post processor for flight data, log files are required to perform the testing.
The first features implemented, concerning the flight data listing, plotting and statistics, can be easily tested thanks to a historical archive of data logs from previous flights. Instead the advance post processing features (parameter identification, frequency and time domain analysis) require specific test points to be performed in flight, for this reason a new mission plan needed to be designed for the autopilot tuning.
A new flight mission must be accurately planned, simulated and validated prior to being performed on a valuable prototype, to ensure a higher rate of success and safety for the mission.

3.1. Test maneuvers

In the early design steps of the advance processing tools, when the scope of tuning the vehicle parameter was set, the necessity to determine the test-manoeuvres to perform in order to identify the correct controller gains from the a/c response arose.

[4]Information of the internal logic architecture of the Piccolo controller are not available from the Cloud Cap Company.

For this purpose a complete list of the vehicle parameters was created to define the effects on the controller outputs, determine if the identification from the flight log was possible and rate the impact of the coefficient modification on the aircraft response. A section of the described list is shown in Figure 7.

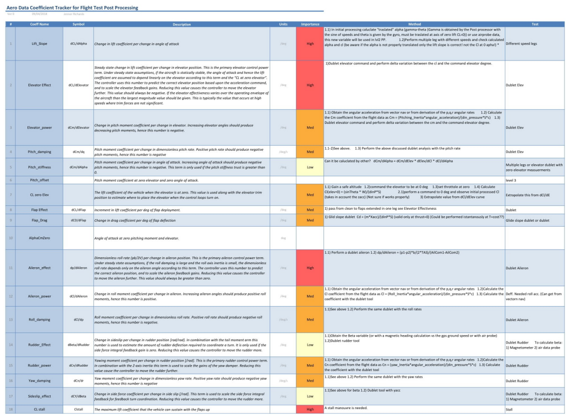


Figure 7: Aero Data Coefficient Tracker for vehicle parameter identification.

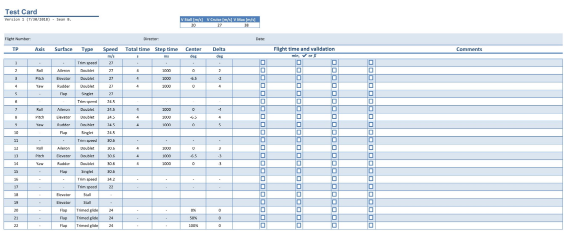
The file was written during an intense research on the controller mechanism, the available data logged and the methods to identify vehicle parameters. It lists all the tuning coefficients of the controller, defining the effects, the level of priority, the possible method to identify the correct value and the test to perform.
The *Aero Data Coefficient Tracker*, which started as a simple table to organize the notes of specific parameters, became an important document for the design of the advance analysis tools and for the autopilot tuning process.

3.2. Mission plan

With the complete scheme of the methods used for the parameter identification, a list of tests to be performed was defined allowing the mission plan to be designed.
Thanks to the fact that the Center for Aerospace Research already had a well structured flight testing activity, it was not required to design a full flight testing plan. Instead the only need was to define the tests to perform, in order to allow the parameter identification and advance optimization. From the Aero Data Coefficient Tracker document it can be noted that 7 different tests were required: 3 main control surfaces doublets, trimmed stable leg, stall, flap singlet and glided slope.
With a defined test list, a full flight mission profile was designed (Figure 8).

Figure 8: Mission plan loaded in Piccolo autopilot.

One of the constrains dictated by the existing flight testing process, is that the test points need be easily executed in a standard flight configuration so that the autopilot parameters can always be analysed and tweaked if necessary, without requiring a dedicated testing process.
After the acceptance of the test plan proposal, a schematic test checklist was required to ease the work for the ground station operator during the flight.
An example of the document is shown in Figure 9.



Test Card
Version 1 (7/30/2018) - Sean B.

V Stall [m/s]	V Cruise [m/s]	V Max [m/s]
20	27	38

Flight Number: Director: Date:

TP	Axis	Surface	Type	Speed	Total time	Step time	Center	Delta	Flight time and validation	Comments
				m/s	s	ms	deg	deg	min, ✔ or ✗	
1	-	-	Trim speed	27	-	-	-	-	☐ ☐ ☐ ☐	
2	Roll	Aileron	Doublet	27	4	1000	0	2	☐ ☐ ☐ ☐	
3	Pitch	Elevator	Doublet	27	4	1000	-6.5	-2	☐ ☐ ☐ ☐	
4	Yaw	Rudder	Doublet	27	4	1000	0	4	☐ ☐ ☐ ☐	
5	-	Flap	Singlet	27					☐ ☐ ☐ ☐	
6	-	-	Trim speed	24.5	-	-	-	-	☐ ☐ ☐ ☐	
7	Roll	Aileron	Doublet	24.5	4	1000	0	-4	☐ ☐ ☐ ☐	
8	Pitch	Elevator	Doublet	24.5	4	1000	-6.5	4	☐ ☐ ☐ ☐	
9	Yaw	Rudder	Doublet	24.5	4	1000	0	5	☐ ☐ ☐ ☐	
10	-	Flap	Singlet	24.5					☐ ☐ ☐ ☐	
11	-	-	Trim speed	30.6	-	-	-	-	☐ ☐ ☐ ☐	
12	Roll	Aileron	Doublet	30.6	4	1000	0	3	☐ ☐ ☐ ☐	
13	Pitch	Elevator	Doublet	30.6	4	1000	-6.5	-3	☐ ☐ ☐ ☐	
14	Yaw	Rudder	Doublet	30.6	4	1000	0	-3	☐ ☐ ☐ ☐	
15	-	Flap	Singlet	30.6					☐ ☐ ☐ ☐	
16	-	-	Trim speed	34.2	-	-	-	-	☐ ☐ ☐ ☐	
17	-	-	Trim speed	22	-	-	-	-	☐ ☐ ☐ ☐	
18	-	Elevator	Stall	-					☐ ☐ ☐ ☐	
19	-	Elevator	Stall	-					☐ ☐ ☐ ☐	
20	-	Flap	Trimed glide	24	-	-	0%	0	☐ ☐ ☐ ☐	
21	-	Flap	Trimed glide	24	-	-	50%	0	☐ ☐ ☐ ☐	
22	-	Flap	Trimed glide	24	-	-	100%	0	☐ ☐ ☐ ☐	

Figure 9: Test card document for autopilot tuning.

A different test card is created for every flight, it contains the list of tests in the temporal sequence with all the parameters to be inserted in the Piccolo Command Center in order to command the maneuvers. Also blank sections are present allowing the Data Processing Assistant to insert time, validation and notes to facilitate the post processing analysis.

4. Results

The designed flight test mission led to the last step of this thesis project: the flight simulation with the Software In the Loop (SIL) technique and real flight tests.

The SIL simulations allowed to test the processing algorithms, prove the parameter identification methods and determine the best coefficients for test points .
Only when the simulation produced an optimized test-card were the tests performed in a real flight. Although the purpose of flight testing was not only to test the software functionality but also to verify the autopilot tuning process, not enough flights were performed with the new mission plan to see a full autopilot tuning process. Some analysis were however made on the data collected in the last flight.
In the following points the major outcomes from the tests are briefly discussed.

- The initial processing was the only feature in the first two parts of the software (*Flight data* and *Flight statistics*) which required a dedicated test to verify its functionality. In fact, while the other functions could be tested with the flight log archive of previous flights, the initial processing required a side to side comparison of the calculated data. This test was performed with the SIL simulations capable to output a complete set of data (not obtainable in real flight) and a reduced log file similar to the one produced by the on-board avionics. The extra data set was then compared with the data calculated in the initial processing algorithm. The results confirmed all the initial processing algorithms, however this test cannot fully validate some of the parameters. In fact the real sensor units are subject to noise, and if this is not properly filtered can be dramatically increased during the derivation, creating erroneous angular accelerations data.

- The effectiveness of the *parameter identification* function was also proven and debugged with extensive simulation tests. The mission plan was repeated several times in different conditions and with different aircraft. The data was then processed with the parameter identification to verify if the values obtained were comparable with the coefficients present in the aerodynamic model of the virtual vehicle simulator. Even if it required several adjustments, at the end of the testing process the feature was capable of identifying all the reported parameters and was finally released.

- The same simulated flight tests containing the control surfaces doublet were then used to test the analysis tools. The objective was to verify if employing the same parameters for the vehicle simulator and for the transfer function simulator, the predicted and the simulated aircraft response matched perfectly. As expected the TFs output and the sensor data collected are slightly different, this is probably due to internal approximations or a simplified version of the equations used in the controller. Nevertheless the discrepancy between the software and the controller prediction is small enough to perform an optimization tuning process, keeping in mind that the maximum precision obtainable is constrained by this discrepant value.

- Lastly the simulation tests were utilized for the characterization of the test points. The listed maneuvers were repeated in different conditions in order to identify the best values for step time, speed, fix stick time, center trim and delta deflection to produce the dynamic response. The excel test card was then fully simulated under the Flight Test Engineer supervision to ensure the safety of the flight before executing it with the real UAV. This process certified the test mission designed, allowing to proceed into a new level of testing.

- The real flight tests immediately highlighted the variable nature of flight conditions which affects the execution of test manoeuvres. While the same test repeated with identical settings and properties in the simulation returns identical values, from the real flight data the identified parameters are different. This result confirms the hypothesis advanced during the design process of the function, which led to implement the averaging feature to extrapolate the mean value from several tests performed. The discrepancy in the values observed expresses the precision limit of the tests. Unlike the parameter identification in wind tunnel tests, in the real flights all the variable conditions occur altering the response. Although the parameters are not uniquely identified, the results obtained demonstrate that the identification of the vehicle coefficients during real flights is possible and it is sensitive enough to perceive variation of the configuration adopted.

- The first notable result from the frequency and time analysis tools, testing with real flights, is the difference between the predicted and the real response of the aircraft. This statement, although it may be perceived as a negative result, is actually confirming the ability of the software to perform a substantial improvement in the autopilot tuning, since the discrepancy in the responses is greater than the margin observed in the simulation between the transfer functions and Piccolo Simulator.

- For several doublet tests the *subspace state-space estimator* revealed the inability to produce an optimized transfer function that replicates the real aircraft response. The algorithm behind the estimator tweaks the polynomial coefficients minimizing the error between the produced transfer function response and the real data. For this reason the outputted TF in most cases does not simulate a realistic response, which lacks typical aircraft behaviour. This phenomena can be more accurately observed with the analysis of the estimated TF in the frequency domain, the gain and phase margins do not assume the typical values of the aircraft dynamics.

- Lastly, during the flight test campaign, the software was employed in the analysis of the steering response related to the mid lane tracking during taxi and T/O. The rolling motion of the a/c is controlled by a track error feedback in which proportional and integral gains contribute to define the output command, which actuates the nose gear and the rudder. This type of analysis required modifications to be applied to the time and frequency input functions. The steering command was imported as a double test to plot the track error with the commanded steering wheel and rudder, studying the response to remove oscillation behaviour. This on-field analysis was possible thanks to the open architecture of the software which allows the user to easily add a temporary function to accomplish a specific task.

The tests planned for the upcoming flights will allow a complete tuning process of a single configuration to be performed, providing the results to verify if the method developed converges at an optimal set of vehicle parameters.

5. Conclusions

At the beginning of this project, the scope was to research a tuning method for the autopilot, optimizing the gain coefficients using the data logs available. This objective rapidly evolved into a more complex project, which was oriented towards creating a versatile and multifunctional tool for the data processing of an automated UAV.

Throughout the thesis it is possible to outline the complete development process of the software, from the initial study of the avionics collecting the data, to the ultimate release and testing of the product generated. As can be perceived in the description of the features, the effectiveness of the development method used was achieved by the instantaneous feedback of the team members using the software. In fact, most of the decisions made are based on the requests advanced in the shared document in order

to generate a software that would fulfil all the individual tasks needed.
The most significant result, is the observable innovation that this project brought to the Center for aerospace research in the field of flight test analysis.
The first improvement, made in the data loading speed time, set the beginning of a new generation of custom data analysis software. Having the ability to read and combine multiple logs along with the implementation of new tools, opened the utilization of the software to other departments of the team who were interested in how the aircraft behaved during flight.
The *statistic data* function integrated allowed the processing of all the historic flight data logs from previous flights, generating a database of general data for each a/c configuration. The data stored in the automatic Excel documents allows to carry out basic trend analysis and mean parameter identification, providing useful information for the design process of systems in new a/c configurations. This tool also created a new system for data sharing within the team.
The group of advanced processing functions (*parameter identification*, *frequency analysis* and *time domain*) supply a complete method for the autopilot tuning of any UAV with similar avionics. More specifically it allows to identify and tweak all the vehicle parameters determining the aircraft response to disturbances or commands which could cause an oscillatory actuator response if not properly set. Even if the objective was not to define a precise effectiveness of the controls but only to give a good estimate in order to reduce the corrective action of the PID controllers, the parameter identification actually gives relatively accurate results and it is therefore recommended to perform all the maneuvers to test specific effectiveness terms, improving the overall aircraft control behaviour.
Although the flight tests performed highlighted some limitations of the software, they also confirmed the tuning methods used and set the basis for future improvements.
With the creation of a new post processing software, not only the data processing work changed, but also the entire data management system evolved. With the new merging and saving functions, after each flight, the fragmented log files are collected and combined into a single Mat file. While the original logs are archived in the server storage, the single file created collects all the data from the flight, improving the organization of the data and helping the team to search for the information needed.
It is easy to understand the positive impact of a customized analysis tool in the context of a flight testing center.

References

[1] Campbell Neil, *The Evolution of Flight Data Analysis*, Proc. Australian Society of Air Safety Investigators conference, 2007.

[2] Cloud Cap Technology, *Piccolo Communications*, v2.2.4, UTC Aerospace Systems, USA, June 23 2015.

[3] P.G. Guest, *Numerical Methods of Curve Fitting: Estimation of Polynomial Coefficients* (Chapter 7), 2012 edition, Cambridge University Press.

[4] Orfanidis, Sophocles J. *Introduction to Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1996.

[5] A. Mornhinweg, *A practical guide to the Piccolo autopilot*, Oklahoma State Unversity, Stillwater (Oklahoma), 2014.

[6] E. Leong, R. Jager, S. Keshmiri and R. Colgren, *Complete Open Loop Dynamics Evaluation of a Small Unmanned Aerial Vehicle Using Different Modeling and Simulation Platforms with Flight Test Validation*, accepted to the proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit, 18-21 Aug 2008, Honolulu, Hawaii.

[7] Dr. Jan Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*, Part I, First printing, DAR Corporation, 1995.

[8] G.F. Franklin, D.J. Powell, and M.L.Workman, *Digital Control of Dynamic Systems*, 3rd Edition, Prentice Hall, 1997.

[9] L. Ljung, *Experiments With Identification of Continuous-Time Models*, Proceedings of the 15th IFAC Symposium on System Identification, 2009.