Escola Tècnica Superior d'Enginyeries
Industrial i Aeronàutica de Terrassa
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Degree: *AEROSPACE ENGINEERING*

Title of the Project:

Performance study of the impacts introduced by new ATM procedures

Contents: Report

Student: Laia Garrigó Invers

Director: Oriol Lizandra Dalmases

Delivery date: 26.09.2014

Acknowledgments

This project has been entirely performed at Advanced Logistic Group (ALG). I am sincerely grateful for the support I have received during the whole development of the project.

I would like to express my gratitude to the people who have helped and supported me throughout the project. I would especially like to thank Andrea Ranieri for his guidance and encouragement throughout these months, as well as Rubén Martínez and Oriol Lizandra, the directors of the project in the enterprise and university respectively.

Finally, I wish to thank my friends and family for their undivided support and unconditional trust.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Aim of the Project

The aim of this project is to develop a flexible software tool dedicated to the assessment of new ATM procedures in terms of aircraft performance.

Chapter 2

Scope of the Project

The development of this project will consist of the following steps:

1. State of the art identification, which includes:
    - Description of the current existing software dedicated to performance analysis;
    - Familiarization with the performance model that will be used for trajectory simulations;
    - Identification and selection of the available programming tools to develop the software.
2. Development of the new software called *TAMS* (Trajectory Analyzer and Modification Software), which can be divided in:
    - Definition of the requirements and capabilities of the tool;
    - Code writing;
    - Code testing and validation report.
3. Application of the new software to a case of analysis, which includes:
    - Definition of the use case and hypotheses
    - Implementation of the software
    - Analysis of the obtained results

Chapter 3

State of the Art

3.1 Background: EUROCONTROL and SESAR

EUROCONTROL is an international organization focused on air traffic management. It was founded in 1960, and it is composed of 38 Member States [1].

Its objective is to promote a uniform Air Traffic Management (ATM) system leading to safer and more economic traffic flow throughout Europe. Four specific Key Performance Areas are currently under special monitoring and optimization according to the European Performance Scheme Regulation (N^o 390/2013): safety, capacity, efficiency, security and environment.

EUROCONTROL is also involved in research, development and validation, and has led the definition phase of the SESAR (Single European Sky ATM Research) project. This is a long-term program formed by almost 300 projects that intends to provide Europe a high-performance ATM infrastructure [2].

In its definition phase, SESAR has defined the European ATM Master Plan: the basis for the new generation of European ATM systems for 2030, a roadmap that will help achieve more sustainable and performing aviation in Europe [3].

The target operational concept of this modernization plan is divided in three steps:

Step 1 Time-based Operations: a synchronized European ATM system focused on flight efficiency and predictability.

Step 2 Trajectory-based Operations: a further-evolved flight efficiency with common 4D trajectory information between partners.

Step 3 Performance-based Operations: the implementation of a European high-performance collaborative ATM system.

3.2 CTA: Controlled Time of Arrival

The objective of the first step is to synchronize trajectory information between Air Traffic Control (ATC) and aircrafts through the Initial 4D Trajectory Management concept.

The purpose is to optimize the arrival traffic at an airport assigning to each aircraft a 2D point named Metering Fix (MF) and a time constraint: the Controlled Time of Arrival (CTA) at this MF. This improves the reliability and accuracy of the arrival sequence.

The procedure consists of a negotiation between ATC and the corresponding aircraft when it is situated at a certain distance from the destination airport. The arrival manager (AMAN) of the airport computes a CTA taking into account the arrival sequences of all other aircrafts and the performance capabilities.

This SESAR concept has been summited to validation mainly through simulations and some flight tests in Europe to demonstrate its technical feasibility under nominal operations [4]. Both methods have provided positive results, and more validation exercises are planned until 2015.

3.3 BADA: Base of Aircraft Data

This section presents BADA, an aircraft performance model designed for use in aircraft trajectory simulations and predictions.

It is developed and maintained by EUROCONTROL through active cooperation with aircraft manufacturers and operating airlines.

BADA can be used for:

- trajectory simulation in the air traffic modelling and simulation tools which are used to support R&D, validation and assessment of new Air Traffic Management (ATM) concepts, Air Traffic Control (ATC) procedures, advanced controller decision support tools and equipment before they are introduced into operational service;
- trajectory prediction in the ground based operational ATM systems (Flight Data Processing Systems) to better plan traffic flows, reduce delays, operating costs and minimize adverse environmental impact, and
- environmental studies in terms of aircraft emissions assessments.

BADA provides two different components: model specifications and datasets.

3.3.1 BADA Model Specifications

Model Specifications are theoretical fundamentals provided in form of generic polynomial expressions. They are used to calculate aircraft performance parameters.

BADA Aircraft Performance Model is based on a kinetic approach to aircraft performance modelling, which models aircraft forces. The motion model that is used within BADA is a so-called Total Energy Model (TEM). TEM equates the rate of work done by forces acting on the aircraft to the rate of increase in potential and kinetic energy. It can be considered as being a reduced point-mass model.

This model provides the necessary equations to calculate performance parameters depending on the engine type, the aircraft configuration, the aircraft mass and the altitude. This section offers an overview of the possibilities that the model offers. For a more detailed description, see *Annex I: User Manual for the Base of Aircraft Data*.

The performance parameters that can be calculated are the following:

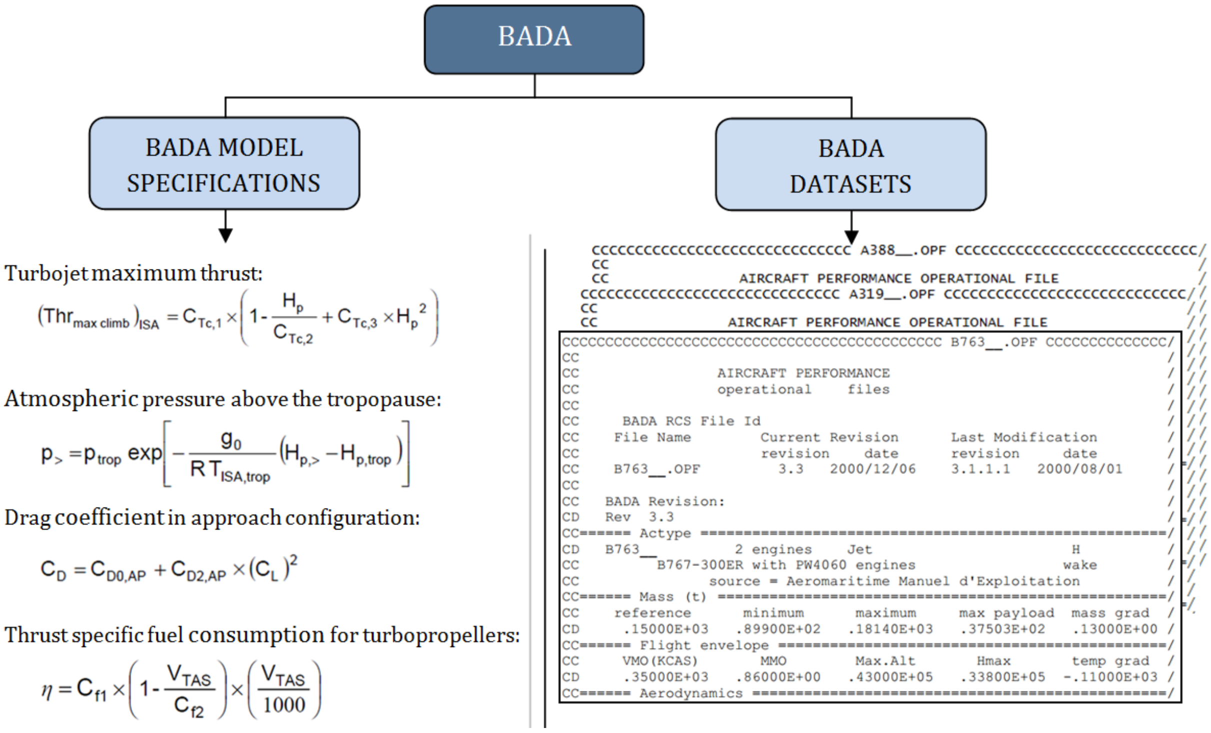
- **Atmospheric properties** (air pressure, temperature, density and speed of sound) as a function of altitude.
- Aircraft **speed schedule** as a function of altitude, aircraft type and flight phase. This schedule is provided in the BADA airline procedure model.
- Aircraft **configuration** (take-off, initial climb, cruise, approach and landing) as a function of the altitude and aircraft speed.
- Aircraft restrictions in terms of **altitude** and **minimum/maximum speed** as a function of aircraft type and mass.
- Aerodynamic variables (**lift and drag**) together with required and available **thrust** for all flight phases and configurations.
- Thrust Specific Fuel Consumption (**TSFC**) and **fuel flow** as a function of all parameters previously commented.

3.3.2 BADA Datasets

The equations mentioned above contain several coefficients characteristics of each aircraft type. A dataset for a given aircraft contains the specific value of this coefficients that particularize the BADA model for an aircraft type.

To examine the format of these files and the different coefficients provided, see *Annex I: User Manual for the Base of Aircraft Data*.

Figure 3.1 shows the two different components that BADA provides. The model equations in terms of general coefficients can be observed on the left, while the corresponding coefficients for a concrete aircraft (in this case a Boeing 767-300ER) are shown on the right.



```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC B763__.OPF CCCCCCCCCCCCCC/
CC                                                                    /
CC                  AIRCRAFT PERFORMANCE                              /
CC                  operational    files                              /
CC                                                                    /
CC     BADA RCS File Id                                               /
CC    File Name         Current Revision      Last Modification       /
CC                      revision    date      revision     date       /
CC    B763__.OPF          3.3   2000/12/06    3.1.1.1   2000/08/01    /
CC                                                                    /
CC   BADA Revision:                                                   /
CD   Rev  3.3                                                         /
CC====== Actype ======================================================/
CD   B763__          2 engines    Jet                     H           /
CC         B767-300ER with PW4060 engines                 wake        /
CC                 source = Aeromaritime Manuel d'Exploitation        /
CC====== Mass (t) ====================================================/
CC    reference      minimum     maximum     max payload  mass grad   /
CD     .15000E+03   .89900E+02   .18140E+03   .37503E+02   .13000E+00 /
CC====== Flight envelope =============================================/
CC     VMO(KCAS)       MMO        Max.Alt       Hmax       temp grad  /
CD     .35000E+03   .86000E+00   .43000E+05   .33800E+05  -.11000E+03 /
CC====== Aerodynamics ================================================/
```

Figure 3.1: Representation of BADA components

The version used in this project will be BADA 3.11, released on May 2013. BADA 3.11 contains aircraft models for 405 different aircraft types. As exposed in [5], these models currently cover 99.85% of the European air traffic as specified by the EUROCONTROL Central Flow Management Unit (CFMU).

These 405 models can be divided into directly supported models and synonym models. In **directly supported models**, aircraft's datasets have been developed using data sources from aircraft manufacturers. This category contains 150 aircrafts. The **synonym models** refer to the other 255 types that are redirected to one of the directly supported aircrafts with similar performance characteristics.

BADA database is usually updated every year, adding coefficient corrections and new models to the database. As mentioned in *Section 5.1: Program Requirements*, upcoming new versions of BADA must be easily updated into *TAMS*.

Figure 3.2 shows BADA traffic coverage in 2012 European air traffic according to EUROCONTROL CFMU.

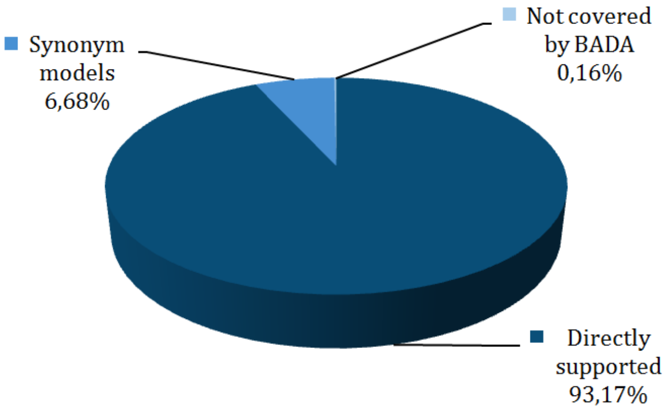


Figure 3.2: European air traffic statistics for 2012

The new software will use BADA as the basis for aircraft performance modelling.

3.4 SAAM: System for traffic Assignment and Analysis at a Macroscopic level

SAAM is an integrated system developed by EUROCONTROL. It is a modelling simulation tool used in the context of Airspace, Network Planning and navigation activities. It is based on BADA as an aircraft performance model.

SAAM's functionalities can be divided into:

- Modelling: for the design of air traffic route networks. 4D trajectories can be generated and applied on any airspace structure.
- Simulation: provides both controller workload and sector capacity analysis for newly designed sectors.

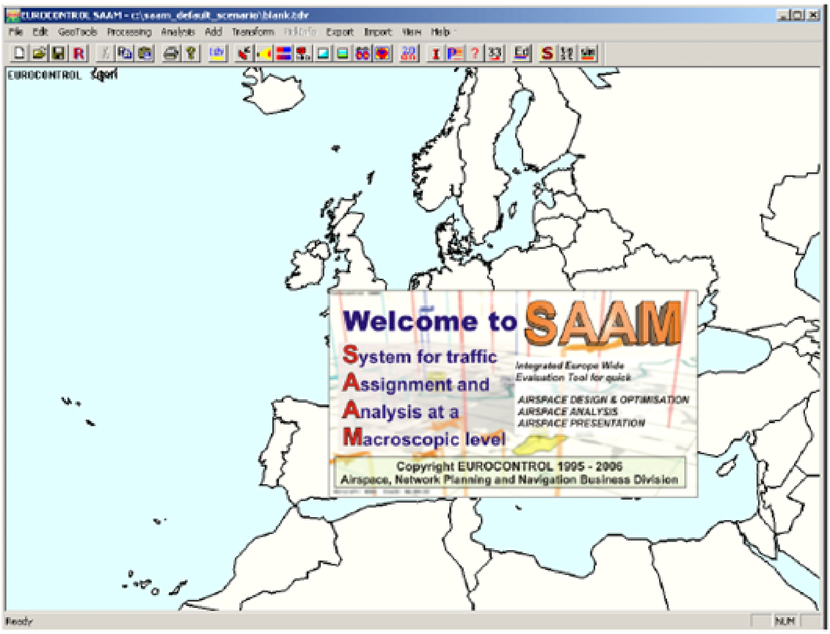


Figure 3.3: SAAM's interface

- Analysis: different sources of data can be selected for analysis and comparisons, using graphs showing variations of airspace load, entry rate, conflict...
- Visualization: SAAM can generate time based animations.

Some of SAAM functionalities interface are shown below.

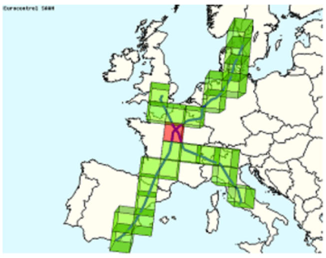


Figure 3.4: SAAM 3D densities

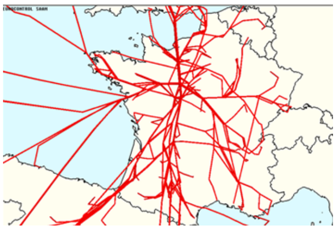


Figure 3.5: SAAM Route Traffic Queries

SAAM is used for the development of this project in two different ways:

- As a *TAMS* validation tool to check analysis results and compare differences,
- To visualize route trajectories and modifications performed by *TAMS*.

us: validacio, criteri, visualitzacio

Chapter 4

Objectives and Motivation of the Project

The objective of this project can be divided in two different parts: the first one is focused on the *development* of a new software tool, and the second one consists of the **application** of this software to assessment and evaluation.

4.1 Tool Development

This project starts with the creation of a new software tool dedicated to performance analysis named, from now on, TAMS (Trajectory Analyzer and Modification Software). This software must be:

flexible the tool must be adaptable to a wide range of uses, compatible with other existing tools, and also prepared for new updates;

reliable the tool must guarantee that the information provided is accurate and validated, identifying its limitations, capabilities and precision, and

user-friendly the software must be simple and efficient, providing an intuitive and attractive interface.

TAMS
Trajectory Analyzer and
Modification Software

4.1.1 Tool advantages compared to existing simulation tools

As seen in *Section 3.4: SAAM: System for traffic Assignment and Analysis at a Macroscopic level*, SAAM is an integrated system used for operational planning. It provides a wide range of functionalities oriented to airspace design, analysis and visualization.

Although its capabilities also include performance analysis and route modelling, this program is focused to perform route network and airspace analysis at a *macroscopic* level. This fact requires to treat with large amounts of data, which is achieved at the expense of accuracy in calculations. SAAM has also restricted to users accredited by EUROCONTROL.

The objective of *TAMS* is to focus in the functionality of route analysis provided by SAAM and enhance it, adding more flexibility to user requirements and adaptability to user inputs, with the improvement of accuracy in the results.

Moreover, *TAMS* will include several functionalities that are not available in SAAM, focused to the further application of the software to new procedures assessment.

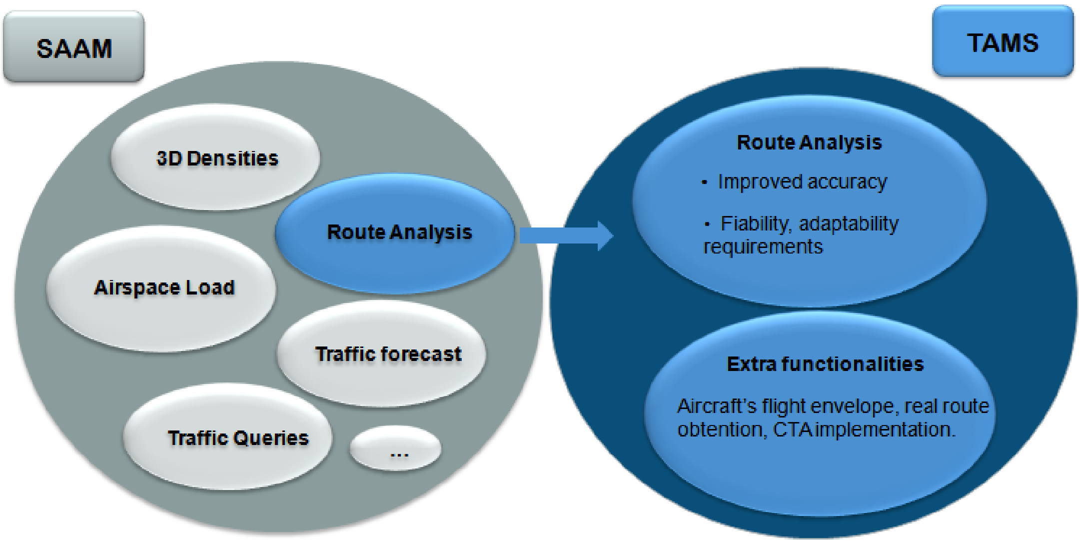


Figure 4.1: Comparison between SAAM and TAMS future functionalities

4.2 Tool Application

Once the tool is developed and validated, the second part of the project consists of the application of the software.

Its purpose is to evaluate the impacts in terms of performance that new procedures would introduce in the current airspace. Not only to analyze these impacts but also to simulate the conditions that the procedures would cause.

More concretely, this study has focused on the effects that the implementation of the CTA would cause at the airport of Barcelona - El Prat.

4.2.1 Tool potential users: ANSPs and Airlines

Modelling and simulation tools are a key enabler to airspace design and optimization. *TAMS* would be used as a support to R&D conducting a quantitative analysis of potential airspace changes. New airspace concepts will be validated before introduced into operational service.

Tool potential users are Air Navigation Service Providers (ANSPs) and airlines, as an assessment for operating planning purposes. *TAMS* will permit to evaluate the performance benefits of new procedures implementation by comparing different scenarios of application.

Part II

Development

Chapter 5

Software Preparation

This chapter will be focused on the steps previous to the development of the program: the software requirements definition and the selection of the tools that will be used to develop it.

5.1 Program Requirements

A general overview of *TAMS* can be observed in Figure 5.1. It will receive user inputs indicating their requirements, and the software will perform the appropriate calculations following BADA model to provide the desired output.

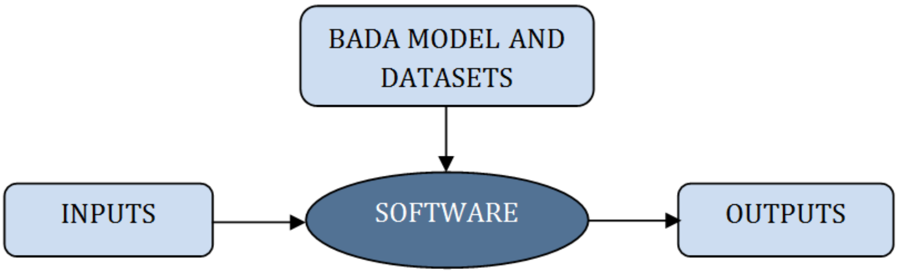


Figure 5.1: General overview of *TAMS* operation.

5.1.1 System capabilities

List of software technical specifications in terms of capabilities.

Identifier	REQ 01.010
Requirement	The program shall be able to calculate the total length and duration of a certain route.

Identifier	REQ 01.020
Requirement	Given the trajectory data of a group of routes, the program shall be able to calculate the fuel consumption associated to each of them and compute the total consumption.

Identifier	REQ 01.030
Requirement	Given the trajectory data of a group of routes, the program shall be able to calculate the aircraft emissions in terms of CO_2 associated to each of them and compute the total consumption.

Identifier	REQ 01.040
Requirement	The program shall be able to identify the flight envelope of each aircraft type, which contains aircraft's limitations in terms of altitude and speed.

Identifier	REQ 01.060
Requirement	The program shall also be able to compute the total difference of a given variable between two different routes. These variables may refer to speed, fuel consumption or aircraft emissions.

Identifier	REQ 01.070
Requirement	The program shall provide the dependencies between a given CTA, the trajectory data (height and remaining distance) and the speed modifications needed in order to accomplish this CTA.

Identifier	REQ 01.080
Requirement	The program shall be able to sequence a group of routes respecting the appropriate separations and aircraft restrictions.

5.1.2 Data processing and interface requirements

List of software technical specifications in terms of data processing.

Identifier	REQ 02.010
Requirement	The system shall be able to read and store operational text files provided by BADA, which contain aircraft characteristics, dimensions and coefficients needed in all functions previously mentioned. These files together occupy about 5 MB of space in the BADA version that will be used (3.11).

Identifier	REQ 02.020
Requirement	The system shall be able to read and process .so6 files, which will contain the trajectory data (aircraft position and timing) needed in the functions mentioned above.

Identifier	REQ 02.030
Requirement	The system shall be prepared to use upcoming new versions of BADA.

Identifier	REQ 02.040
Requirement	The system shall display the results in text files when required, e.g. a text file with the routes resulted from the program modifications, or a text file with all performance parameters of a specific route.

Identifier	REQ 02.050
Requirement	The system shall provide an attractive and intuitive interface, allowing the user to get access quickly to the different use cases and scenarios.

5.2 Software Environment

Once the requirements are established, it is possible to choose the appropriate tools that will be used to fulfill them. Since the main objective of this part is to develop new software, it is necessary to determine two aspects [6]:

the programming language , which is used to write a series of human understandable computer instructions that can be read and translated into machine code, and

the Integrated Development Environment (IDE) , which brings all the programmer tools that are necessary to create a program (editor, compiler, linker and debugger).

5.2.1 Programming language selection

A large number of programming languages exist nowadays (see [7]). Two different candidates have been studied:

C++ Language

C++ is a middle-level programming language developed by Bjarne Stroustrup starting in 1979. It was oriented to systems programming with performance, efficiency and flexibility of use as its design requirements. C++ has also been found useful in many other contexts, including desktop applications and entertainment software, such as video games [8].

According to [8, 9], the advantages and disadvantages of this language can be summarized in:

Advantages:

- It is a compiled language, since it is translated to the machine‘s native language by a program called compiler. This factor makes C++ a really fast language.
- Because of its object-oriented structure, it is well-suited for large projects since the code can be easily reused.
- It offers an large library support.

Disadvantages:

- Since C++ is a very broad language used for very different applications, it has a more complicated syntax compared to other languages.

VBA Language

Visual BASIC for Applications (VBA), is a computer programming language developed by Microsoft. It allows the development of user-defined functions and the automation of certain processes and calculations.

VBA is nowadays a standard feature of Microsoft Office products, and it allows the user to create structured programs directly in these Office products, such as Excel, Word and Power Point [10].

According to [9, 10], the advantages and disadvantages of this language can be summarized in:

Advantages:

- It is a language relatively easy to learn, user-friendly and facilitates the use of visual applications.

- The final product can be distributed only by copying the original file where it has been created, such as the Excel file.
- It offers an large library support.

Disadvantages:

- It is an interpreted language. As opposed to compiled languages, interpreted languages are directly read and executed by the interpreter. This process leads to run slower and less efficiently.
- The final product is not a stand-alone program, it is attached to the Excel file where it has been created.
- Moreover, this dependence can lead to incompatibilities between Excel versions.
- Its simplicity leads to more limited functions and possibilities compared to C++.

For all these reasons previously exposed, the programming language selected for this project is **C++**.

5.2.2 IDE Selection

Once the programming language is selected, it is necessary to choose the appropriate IDE compatible with this language. Two candidates have been compared: Qt Creator and Microsoft Visual C++.

Qt and its supporting tools are developed as an open source project. It is also a cross-platform application, which means that can operate on multiple computer platforms and operating systems such as Windows, Mac OS X or Linux [11].

In this case, the decision has not been as immediate as compared to language selection. Both environments are powerful, flexible IDEs adapted to GUI[1]programming [11, 13].

However, open source software and cross-platform adaptability (and also personal preference) have made Qt Creator the Integrated Development Environment chosen for this project.

Figure 5.2: Qt and its cross-platform user experience can connect its applications to any operating system and devices.

---

[1] Graphical User Interface, a type of interface that allows users to interact with a program through graphical icons and visual indicators[12].

Chapter 6

Software Functionalities

6.1 Functions Classification

The functionalities of *TAMS* can be divided in three different blocks, as shown in Figure 6.1. The first one is based on *route and aircraft analysis*, either to obtain the characteristics of a given route or the limitations of a particular aircraft.

The second block is focused on *route modifications*. As its name implies, it adjusts the trajectory of a group of routes following the requirements introduced by the user. These modifications have sequencing and merging purposes, and they have been created for the Use Case application.

Finally, *TAMS* has one extra function called DataDownload Function. Its objective is to download actual route trajectories to create new traffic files.

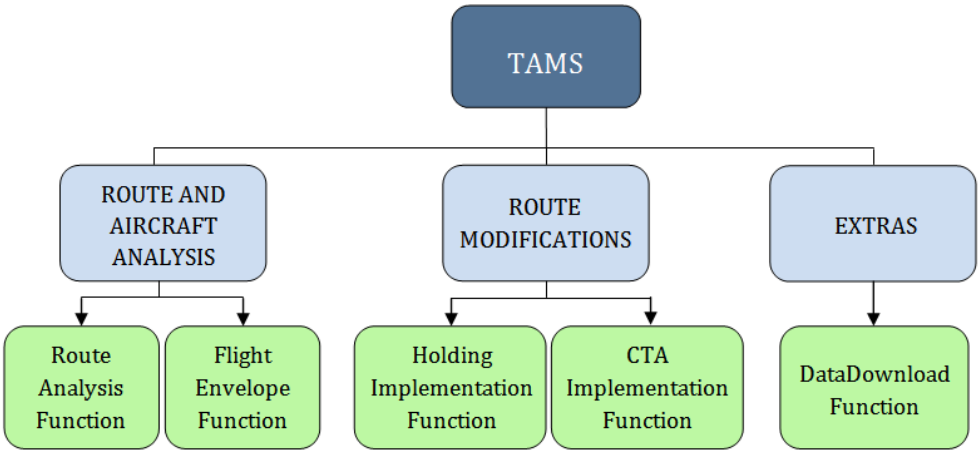


Figure 6.1: Diagram of *TAMS* blocks and functions

6.2 Software input: Traffic Files

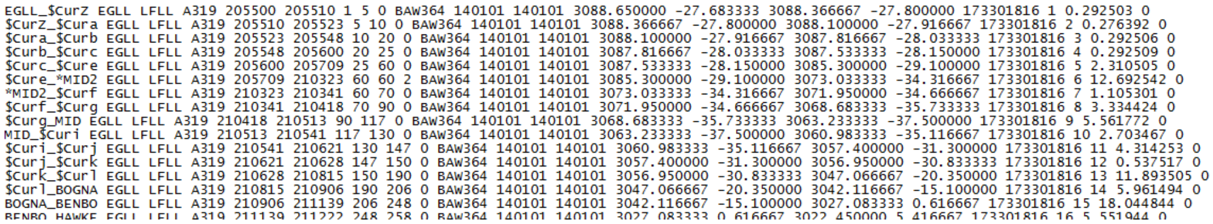
The main input of *TAMS* functions are traffic files. These files contain information about the trajectory of single route or group of routes in a text format.

EUROCONTROL has developed the DDR Project (Demand Data Repository Project), which provides airspace planners and airspace users the past European traffic demand obtained from the CFMU, among other features. All European flights from 2011 up to now are available for registered users acredited by EUROCONTROL. For further details of this project, see [14].

The route information provided in these files consists of: origin and destination of the flight, route aircraft and callsign together with the route trajectory segmented in several parts. For each segment, the following data is presented:

- UTC time and date of the beginning and the end of the segment.
- Flight level begin and end.
- Corresponding latitude and longitude coordinates.

This information is stored in a traffic file: a text file with *.so6* extension. Figure 6.2 shows a traffic file example of flight KAC102, flying from Heathrow Airport (London) to Lyon Saint Exupery Airport (France) on January 1^{st} 2014.



```
EGLL_$CurZ EGLL LFLL A319 205500 205510 1 5 0 BAW364 140101 140101 3088.650000 -27.683333 3088.366667 -27.800000 173301816 1 0.292503 0
$CurZ_$Cura EGLL LFLL A319 205510 205523 5 10 0 BAW364 140101 140101 3088.366667 -27.800000 3088.100000 -27.916667 173301816 2 0.276392 0
$Cura_$Curb EGLL LFLL A319 205523 205548 10 20 0 BAW364 140101 140101 3088.100000 -27.916667 3087.816667 -28.033333 173301816 3 0.292506 0
$Curb_$Curc EGLL LFLL A319 205548 205600 20 25 0 BAW364 140101 140101 3087.816667 -28.033333 3087.533333 -28.150000 173301816 4 0.292509 0
$Curc_$Cure EGLL LFLL A319 205600 205709 25 60 0 BAW364 140101 140101 3087.533333 -28.150000 3085.300000 -29.100000 173301816 5 2.310505 0
$Cure_*MID2 EGLL LFLL A319 205709 210323 60 60 2 BAW364 140101 140101 3085.300000 -29.100000 3073.033333 -34.316667 173301816 6 12.692542 0
*MID2_$Curf EGLL LFLL A319 210323 210341 60 70 0 BAW364 140101 140101 3073.033333 -34.316667 3071.950000 -34.666667 173301816 7 1.105301 0
$Curf_$Curg EGLL LFLL A319 210341 210418 70 90 0 BAW364 140101 140101 3071.950000 -34.666667 3068.683333 -35.733333 173301816 8 3.334424 0
$Curg_MID EGLL LFLL A319 210418 210513 90 117 0 BAW364 140101 140101 3068.683333 -35.733333 3063.233333 -37.500000 173301816 9 5.561772 0
MID_$Curi EGLL LFLL A319 210513 210541 117 130 0 BAW364 140101 140101 3063.233333 -37.500000 3060.983333 -35.116667 173301816 10 2.703467 0
$Curi_$Curj EGLL LFLL A319 210541 210621 130 147 0 BAW364 140101 140101 3060.983333 -35.116667 3057.400000 -31.300000 173301816 11 4.314253 0
$Curj_$Curk EGLL LFLL A319 210621 210628 147 150 0 BAW364 140101 140101 3057.400000 -31.300000 3056.950000 -30.833333 173301816 12 0.537517 0
$Curk_$Curl EGLL LFLL A319 210628 210815 150 190 0 BAW364 140101 140101 3056.950000 -30.833333 3047.066667 -20.350000 173301816 13 11.893505 0
$Curl_BOGNA EGLL LFLL A319 210815 210906 190 206 0 BAW364 140101 140101 3047.066667 -20.350000 3042.116667 -15.100000 173301816 14 5.961494 0
BOGNA_BENBO EGLL LFLL A319 210906 211139 206 248 0 BAW364 140101 140101 3042.116667 -15.100000 3027.083333 0.616667 173301816 15 18.044844 0
BENBO_HAWKE EGLL LFLL A319 211139 211222 248 258 0 BAW364 140101 140101 3027.083333 0.616667 3022.450000 5.416667 173301816 16 5.551944 0
```

Figure 6.2: Example of the first lines of a traffic file

Each line is composed of 20 columns. The information contained in each line is shown in Table 6.1.

Two different types of traffic files can be found in the DDR database: "m1" traffic files and "m3" traffic files.

- **m1 traffic files** contain the last filled flight plan trajectory of the route. It corresponds to the scheduled flight, the route that the aircraft should follow established previous to its departure.
- **m3 traffic files** contain the flight plan trajectory enhanced with radar data. As a consequence, it represents a more realistic route.

0	Segment name	**10**	Date begin (yymmdd)
1	Origin	**11**	Date end (yymmdd)
2	Destination	**12**	Latitude begin (minute decimale)
3	Aircraft type (ICAO code)	**13**	Longitude end (minute decimale)
4	Time begin segment (hhmmss)	**14**	Latitude end (minute decimale)
5	Time end segment (hhmmss)	**15**	Longitude end (minute decimale)
6	Flight level begin segment	**16**	Flight ID
7	Flight level end segment	**17**	Sequence (line number)
8	Status	**18**	Segment length (NM)
9	Callsign	**19**	Color

Table 6.1: Components of a single traffic file line

Traffic files are also used in other modelling simulation programs such as System for traffic Assignment & Analysis at Macroscopic level (SAAM) to model, analyze & visualize route network. Figure 6.3 shows both m1 and m3 files of the flight KAC102 using SAAM to display them.

Figure 6.3: Route visualization of the m1 file (red) and m3 file (purple) of flight KAC102.

With this information it is possible to describe the 4D trajectory of a route. This files will be used by *TAMS* to analyze routes and modify them.

6.3 Functions Description

6.3.1 Route Analysis Function

Given a traffic file containing route trajectory data (see *Section 6.2: Software input: Traffic Files*), it is possible to obtain route's main characteristics: its length, duration, consumed fuel, and the carbon dioxide (CO_2) emitted during fuel combustion.

This analysis allows comparing two routes quantitatively, and also to evaluate the modifications made on a single route in terms of time reduction or fuel consumption. The *Route Analysis* function is the main function of *TAMS*, and it is used as a tool to reflect the direct benefits of the new procedure's implementation.

General procedure

The method used in *TAMS* to analyze a route will be called, from now on, as *Step by Step*. This method can be summarized in three parts:

1. Route discretization: division of the route in several segments.
2. For each segment, calculation of the required variables (length, time, fuel and CO_2) based on BADA model.
3. Once the last segment has been analyzed, calculation of the final variables taking into account the contributions of each segment.

This method can be summarized in Figure 6.4 as an example to obtain the consumed fuel of a route. Detailed explanation of the method can be found in *Annex II: TAMS Functions*.

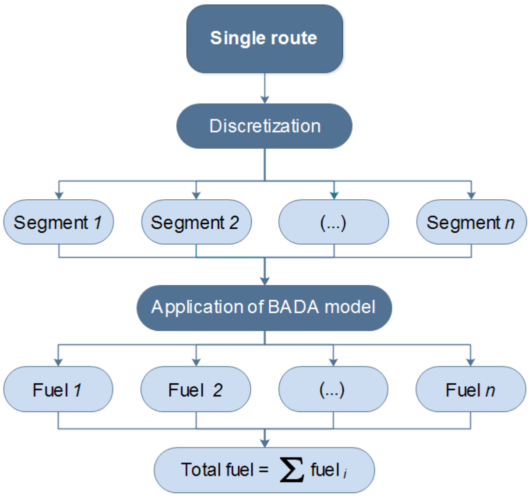


Figure 6.4: Flow chart of Route Analysis Function

Interface and options

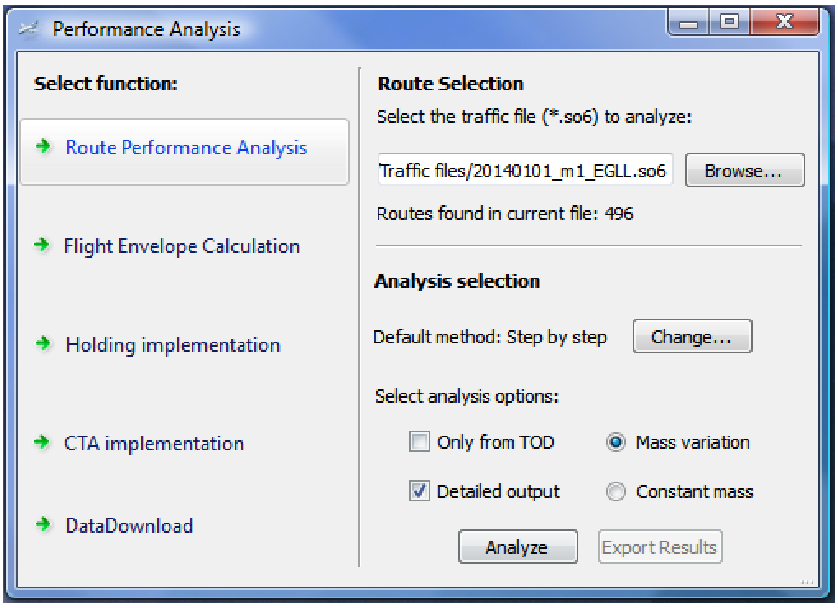
The Route Analysis function is flexible and adaptable to user requirements. As seen in Figure 6.5, several options are available once the traffic file is loaded.

First option corresponds to the analysis method. If desired, it is possible to use another method to analyze routes, such as the method used by SAAM. This option is implemented only to validate *TAMS* analysis (see *Chapter 7: Software Validation*).

Another option used for validation purposes is the possibility to maintain aircraft mass constant during the analysis, or rather take into account mass variation caused by the consumed fuel.

It is also possible to analyze a route only from Top Of Descent (TOD). In this case, *TAMS* omits the climb and cruise parts of the flight and only evaluates aircraft performance from TOD to landing. This option is used in the Use Case analysis (see *Chapter 8: Use Case Definition*).

Finally, the output generated by *TAMS* can be either generic or detailed. This option is explained further in this section.



Route Selection

Select the traffic file (*.so6) to analyze: Traffic files/20140101_m1_EGLL.so6

Routes found in current file: 496

Analysis selection

Default method: Step by step

Select analysis options:

☐ Only from TOD ● Mass variation

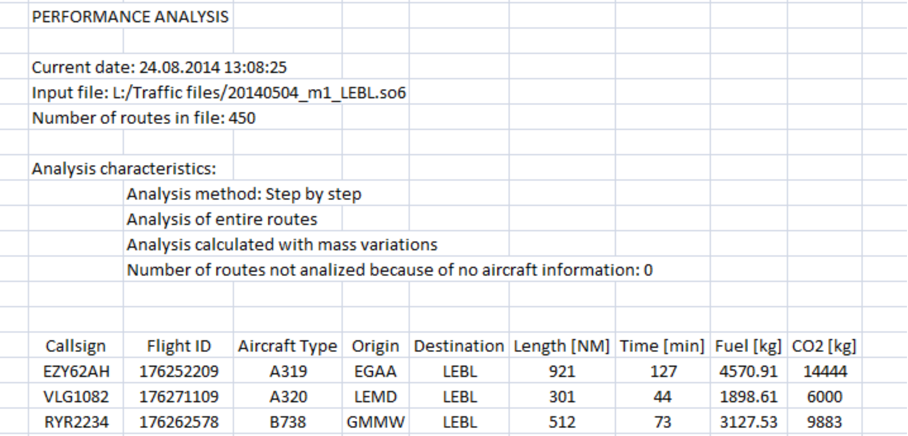
☑ Detailed output ○ Constant mass

Figure 6.5: Route Analysis Function interface

Function outputs

Once the analysis is concluded, *TAMS* generates a text file with the results of the analysis as shown in Figure 6.6. The information provided consists of:

- Date and time of the analysis
- Name of the traffic file analyzed with the total routes contained in it
- Method and characteristics of the analysis
- List of flights with corresponding data and results



PERFORMANCE ANALYSIS

Current date: 24.08.2014 13:08:25
Input file: L:/Traffic files/20140504_m1_LEBL.so6
Number of routes in file: 450

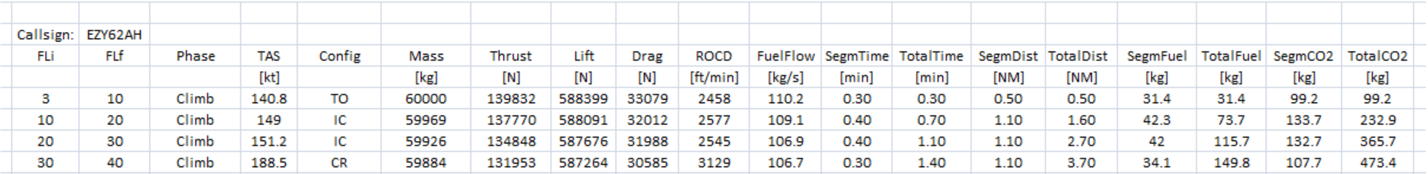
Analysis characteristics:
Analysis method: Step by step
Analysis of entire routes
Analysis calculated with mass variations
Number of routes not analized because of no aircraft information: 0

Callsign	Flight ID	Aircraft Type	Origin	Destination	Length [NM]	Time [min]	Fuel [kg]	CO2 [kg]
EZY62AH	176252209	A319	EGAA	LEBL	921	127	4570.91	14444
VLG1082	176271109	A320	LEMD	LEBL	301	44	1898.61	6000
RYR2234	176262578	B738	GMMW	LEBL	512	73	3127.53	9883

Figure 6.6: Output of the Route Analysis Function

If desired, it is possible to obtain a detailed analysis of each route (see Figure 6.7), containing the following information for each segment of the route:

- Initial and final flight levels
- Segment phase and configuration
- Aircraft mass and speed (TAS)
- Thrust, lift, drag, ROCD and fuel flow
- Segment and total time, distance, fuel and CO_2



Callsign: EZY62AH

FLi	FLf	Phase	TAS [kt]	Config	Mass [kg]	Thrust [N]	Lift [N]	Drag [N]	ROCD [ft/min]	FuelFlow [kg/s]	SegmTime [min]	TotalTime [min]	SegmDist [NM]	TotalDist [NM]	SegmFuel [kg]	TotalFuel [kg]	SegmCO2 [kg]	TotalCO2 [kg]
3	10	Climb	140.8	TO	60000	139832	588399	33079	2458	110.2	0.30	0.30	0.50	0.50	31.4	31.4	99.2	99.2
10	20	Climb	149	IC	59969	137770	588091	32012	2577	109.1	0.40	0.70	1.10	1.60	42.3	73.7	133.7	232.9
20	30	Climb	151.2	IC	59926	134848	587676	31988	2545	106.9	0.40	1.10	1.10	2.70	42	115.7	132.7	365.7
30	40	Climb	188.5	CR	59884	131953	587264	30585	3129	106.7	0.30	1.40	1.10	3.70	34.1	149.8	107.7	473.4

Figure 6.7: Detailed output of the Route Analysis Function for flight EZY62AH

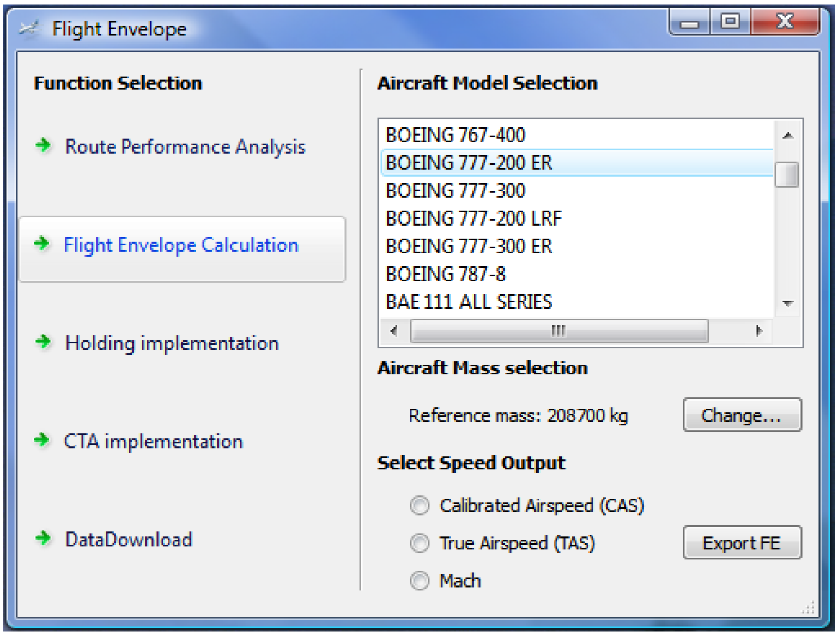
6.3.2 Flight Envelope Function

The flight envelope of an aircraft defines the combinations of speed and altitude at which the aircraft can fly without exceeding its limitations [15].

On the one hand, the Flight Envelope Function can be used for informational purposes. Given a certain aircraft with its corresponding mass, it is possible to obtain its flight envelope. The user has to select:

- The aircraft of analysis. All BADA models are loaded and displayed on a list.
- If desired, it is possible to change the reference mass that BADA assigns to each aircraft by any other mass inside aircraft capabilities.
- The required output speed: it can be shown in terms of CAS, TAS or Mach (see *Annex I, Section 3.1* to know the differences between speeds).

The interface of this function is shown in Figure 6.8.



BOEING 767-400
BOEING 777-200 ER
BOEING 777-300
BOEING 777-200 LRF
BOEING 777-300 ER
BOEING 787-8
BAE 111 ALL SERIES

Reference mass: 208700 kg

Figure 6.8: Flight Envelope Function interface

On the other hand, this function is also used as a controller in the rest of *TAMS* functions. When the software has to modify a route or assign the aircraft speed at a particular point in time, it must be certain that the aircraft is flying inside its performance limitations.

General procedure

The procedure to obtain the flight envelope is summarized in Figure 6.9. Detailed explanation can be found in *Annex II: TAMS Functions*.

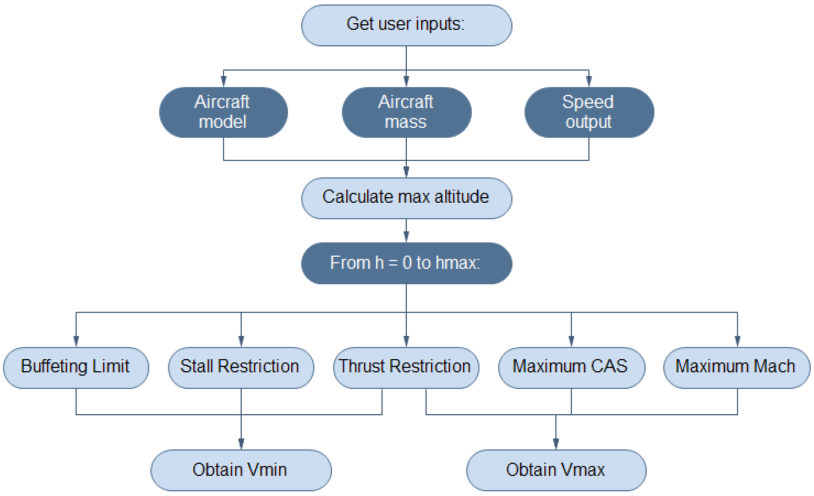


Figure 6.9: Flow chart of Flight Envelope Function

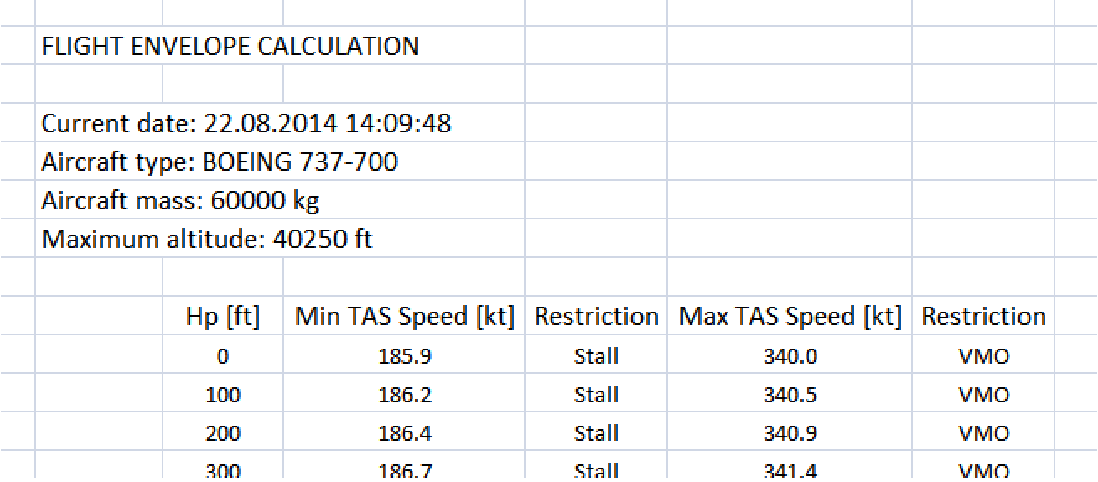
This analysis consists on calculating, for each altitude, the minimum and maximum speeds of the aircraft. These speeds are found applying five different restrictions on aircraft performance:

1. Thrust Restriction: Aircraft speed is limited by the available engine thrust. This limitation occurs when the maximum thrust available equals the required thrust.
2. Stall Restriction: Stall occurs when the wing can not provide enough lift to maintain the aircraft in level flight, which may occur at low speed.
3. Low Speed Buffeting Limit: Aeroelastic effect that can appear because of airflow separation exciting some parts of the aircraft.
4. Maximum operating speeds: constant CAS speed and Mach that should not be overcame to ensure aircraft operability. Provided by the aircraft manufacturer.
5. Maximum altitude: There is a certain altitude where the lift generated by the wing can not overcome aircraft weight to increase flight level, and an increment in speed no longer results in an altitude raise.

Each restriction provides a candidate for the minimum and/or maximum speed. In the case of minimum speed, the final result will be the maximum of the candidates, since it is the most restrictive. Same procedure with maximum speed.

Function outputs

The flight envelope of the aircraft is provided in a text file. Figure 6.10 corresponds to the flight envelope of a Boeing 737-700 in TAS speed. The corresponding graphic is shown in Figure 6.11, with the most limiting restriction at each altitude.



FLIGHT ENVELOPE CALCULATION

Current date: 22.08.2014 14:09:48
Aircraft type: BOEING 737-700
Aircraft mass: 60000 kg
Maximum altitude: 40250 ft

Hp [ft]	Min TAS Speed [kt]	Restriction	Max TAS Speed [kt]	Restriction
0	185.9	Stall	340.0	VMO
100	186.2	Stall	340.5	VMO
200	186.4	Stall	340.9	VMO
300	186.7	Stall	341.4	VMO

Figure 6.10: Flight Envelope Function output

This aircraft can fly until 40250 ft. At low altitudes, the stall speed is the dominant restriction for Vmin. Since CAS stall speed is constant with altitude, TAS airspeed increases at the aircraft gets higher. From 15000 ft to its maximum altitude, speed is limited by buffeting.

Referring to maximum speed, at low altitudes the aircraft is limited by its maximum operating speed (V_{MO}). This value is a constant CAS that implies increasing TAS with altitude, since air density decreases. Above 26000 ft, maximum operating Mach number (M_{MO}) becomes the dominant restriction. From 26000 ft to 36000 ft (below tropopause) constant M_{MO} implies decreasing TAS with altitude. Above tropopause, since temperature remains constant, TAS has the same behaviour.

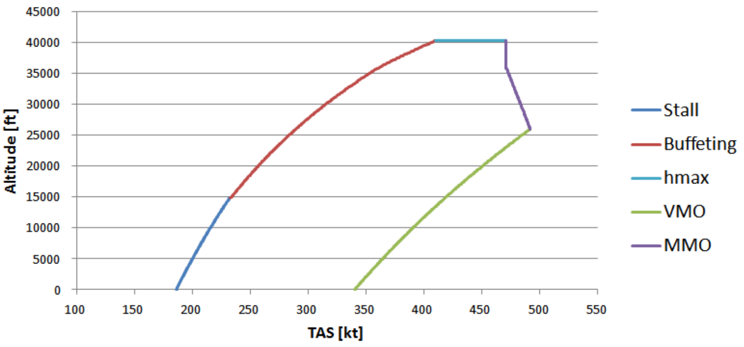


Figure 6.11: Boeing 737-700 flight envelope representation

6.3.3 Holding Implementation Function

This function is used to simulate the second scenario of the Use Case. Given a group of routes, this scenario simulates how all trajectories would be if the delay was absorbed exclusively by holdings. The justification and hypotheses taken can be found in *Section 8: Use Case Definition*.

General procedure

The procedure to apply the holding function is summarized in Figure 6.12.

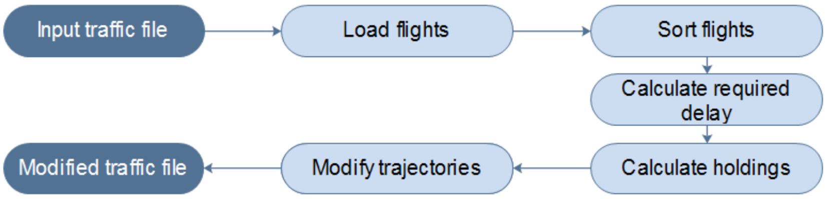


Figure 6.12: Flow chart of Holding Implementation Function

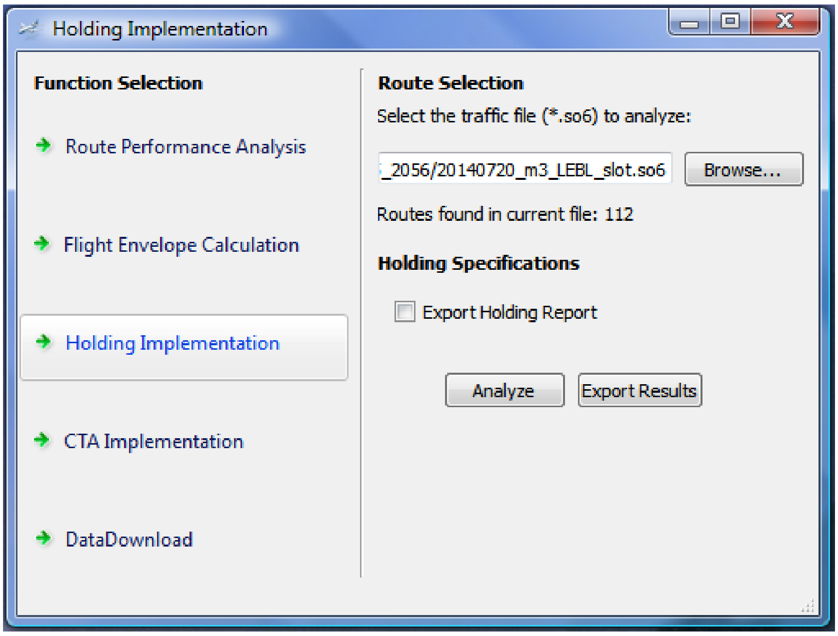
The general procedure starts with the original traffic file that the user introduces. Flights appearing in this file are sorted by timing and queued. The program continues calculating the separations between each flight and the required delay to accomplish them.

With a delay associated to each aircraft, the calculation to the corresponding holdings is straightforward. The original trajectory is finally modified.

Interface and options

The interface of the Holding Implementation Function in shown in Figure 6.13.

In addition to the traffic file with the modified trajectories, it is also possible to export a text file named Holding Report with the delays and holdings (explained in next section).



Select the traffic file (*.so6) to analyze: _2056/20140720_m3_LEBL_slot.so6

Routes found in current file: 112

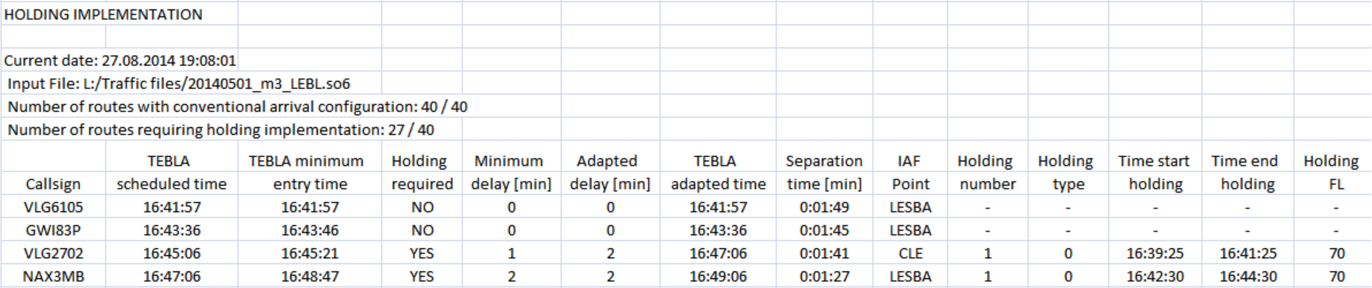
☐ Export Holding Report

Figure 6.13: Holding Implementation Function interface

Function outputs

Once the calculations are done, *TAMS* provides two different files. The first one corresponds to the modified traffic file with the necessary holdings inserted to the routes.

The second file is named Holding Report and it contains a list of each flight with the delays and holdings associated to each flight as shown in Figure 6.14.



HOLDING IMPLEMENTATION

Current date: 27.08.2014 19:08:01
Input File: L:/Traffic files/20140501_m3_LEBL.so6
Number of routes with conventional arrival configuration: 40 / 40
Number of routes requiring holding implementation: 27 / 40

Callsign	TEBLA scheduled time	TEBLA minimum entry time	Holding required	Minimum delay [min]	Adapted delay [min]	TEBLA adapted time	Separation time [min]	IAF Point	Holding number	Holding type	Time start holding	Time end holding	Holding FL
VLG6105	16:41:57	16:41:57	NO	0	0	16:41:57	0:01:49	LESBA	-	-	-	-	-
GWI83P	16:43:36	16:43:46	NO	0	0	16:43:36	0:01:45	LESBA	-	-	-	-	-
VLG2702	16:45:06	16:45:21	YES	1	2	16:47:06	0:01:41	CLE	1	0	16:39:25	16:41:25	70
NAX3MB	16:47:06	16:48:47	YES	2	2	16:49:06	0:01:27	LESBA	1	0	16:42:30	16:44:30	70

Figure 6.14: Holding Report example

6.3.4 CTA Implementation Function

This function is used to simulate the third scenario of the Use Case. Given a group of routes, this scenario simulates how all trajectories would be if the delay was absorbed sequencing by CTA. The use of this function and hypotheses taken can be found in *Section 8: Use Case Definition.*

General procedure

The procedure to apply the CTA function is summarized in Figure 6.15.

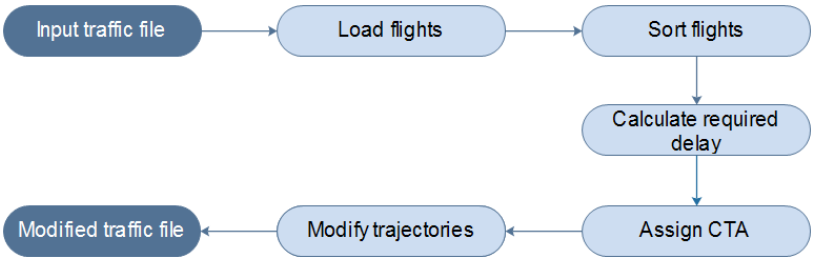


Figure 6.15: Flow chart of CTA Function

The procedure is quite similar to the previous one. Flights appearing in the input traffic file are sorted by timing and queued. The program continues calculating the separations between each flight and the required delay to accomplish them.

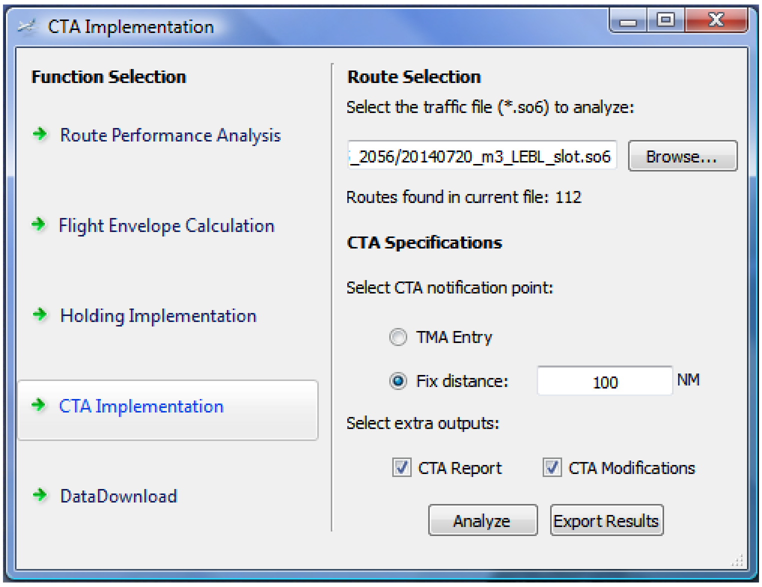
The difference between the Holding Implementation function and this one is the modification done to each route to accomplish its CTA. *TAMS* modifies aircraft speed from the moment its CTA has been notified to the arriving point.

Interface and options

Once the traffic file is loaded, *TAMS* enables several options to implement CTA.

The notification point of the CTA (i.e. the point of the route where the aircraft is informed of its CTA) can either be when the aircraft enters Barcelona TMA or when it is situated at a fix distance to the airport.

Besides the output traffic file, the user can also export two different text files with detailed information of the simulation. These documents are explained in next section.



CTA Implementation

Function Selection

Route Selection

Select the traffic file (*.so6) to analyze:

_2056/20140720_m3_LEBL_slot.so6 Browse...

Routes found in current file: 112

CTA Specifications

Select CTA notification point:

○ TMA Entry

● Fix distance: 100 NM

Select extra outputs:

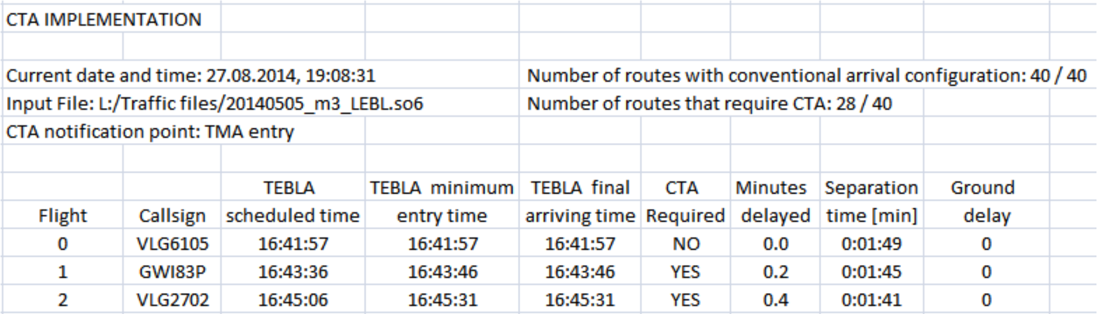
☑ CTA Report ☑ CTA Modifications

Figure 6.16: CTA Implementation Function interface

Function outputs

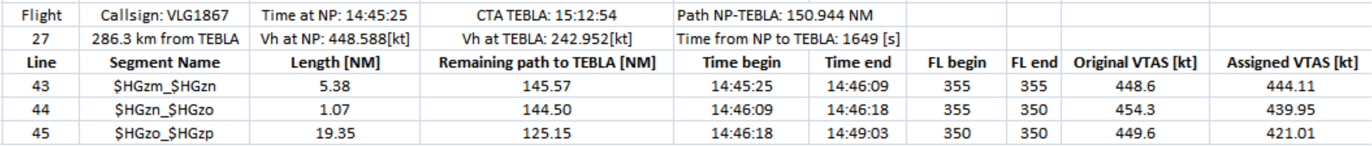
Once the calculations are done, *TAMS* provides three different files:

- The first one corresponds to traffic file with the modified route trajectories.
- The second file is named CTA Report and it contains a list of each flight with the delays and CTA associated to each flight as shown in Figure 6.17.
- Finally, a text file named CTA Modifications contains detailed information of requirements and modifications done for each segment of the route (Figure 6.18).



CTA IMPLEMENTATION								
Current date and time: 27.08.2014, 19:08:31			Number of routes with conventional arrival configuration: 40 / 40					
Input File: L:/Traffic files/20140505_m3_LEBL.so6			Number of routes that require CTA: 28 / 40					
CTA notification point: TMA entry								
		TEBLA	TEBLA minimum	TEBLA final	CTA	Minutes	Separation	Ground
Flight	Callsign	scheduled time	entry time	arriving time	Required	delayed	time [min]	delay
0	VLG6105	16:41:57	16:41:57	16:41:57	NO	0.0	0:01:49	0
1	GWI83P	16:43:36	16:43:46	16:43:46	YES	0.2	0:01:45	0
2	VLG2702	16:45:06	16:45:31	16:45:31	YES	0.4	0:01:41	0

Figure 6.17: CTA Report example



Flight	Callsign: VLG1867	Time at NP: 14:45:25	CTA TEBLA: 15:12:54	Path NP-TEBLA: 150.944 NM					
27	286.3 km from TEBLA	Vh at NP: 448.588[kt]	Vh at TEBLA: 242.952[kt]	Time from NP to TEBLA: 1649 [s]					
Line	**Segment Name**	**Length [NM]**	**Remaining path to TEBLA [NM]**	**Time begin**	**Time end**	**FL begin**	**FL end**	**Original VTAS [kt]**	**Assigned VTAS [kt]**
43	$HGzm_$HGzn	5.38	145.57	14:45:25	14:46:09	355	355	448.6	444.11
44	$HGzn_$HGzo	1.07	144.50	14:46:09	14:46:18	355	350	454.3	439.95
45	$HGzo_$HGzp	19.35	125.15	14:46:18	14:49:03	350	350	449.6	421.01

Figure 6.18: CTA Modifications example for one single flight

6.3.5 DataDownload Function

This function is to create new traffic files with real-time trajectories. The necessary information to create them is obtained from FlightRadar24, a website that provides real-time information about flights that are airborne.

The necessity of this function together with its use in the Use Case and more details, can be found in *Section 8: Use Case Definition.*

General procedure

This function connects *TAMS* to FlightRadar24 website, and downloads the information displayed every certain period of time. The procedure to obtain this information is shown in Figure 6.19.

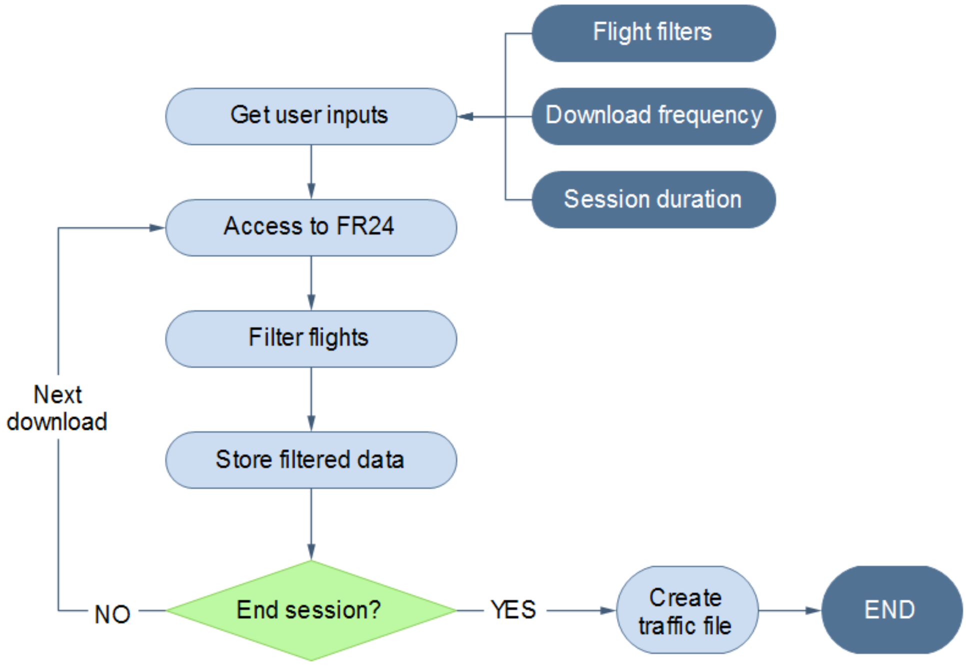


Figure 6.19: Flow chart of the DataDownload Function

It consists of:

1. The user introduces its inputs to *TAMS*:
   - the flights he wants to download (e.g. flights flying to Barcelona-El Prat)
   - the download frequency (e.g. 15 seconds): at every download, *TAMS* obtains the position and information displayed in the website. Higher frequencies

lead to more precise traffic files.

  - the total duration of the session (e.g. 4 hours).

2. Start of first download: *TAMS* accesses to FlightRadar24 website and gets a list of all flights that are airborne at that moment.
3. *TAMS* filters the flights according to user inputs.
4. The information of the flights that accomplish user filters is stored.
5. If the 4 hour period has elapsed, create the traffic file with all information downloaded during the session. Else, start a new download after the 15 seconds of the example.

Interface

Figure 6.20 shows the interface of DataDownload Function while running:

```
/**********************************************************************/
         FLIGHT INFORMATION DOWNLOAD FROM WWW.FLIGHTRADAR24.COM
/**********************************************************************/

 Indicate the interval between downloads (seconds): 15

 Indicate the total duration of the download (minutes): 240

 Proxy identification required? (Y/N): N


 Session started. Total downloads required:  960

 Current download filters:


  Aircrafts flying to BCN airport


 Download number  1

   Current time:  11:42:26

   Accessing web data...

   Web Data downloaded successfully.

   Flights with required filters:  30

 Download number  2

   Current time:  11:42:41

   Accessing web data...

   Web Data downloaded successfully.

   Flights with required filters:  30

 Download number  3

   Current time:  11:42:56

   Accessing web data...

   Web Data downloaded successfully.

   Flights with required filters:  29

 Download number  4

   Current time:  11:43:11

   Accessing web data...
```

Figure 6.20: DataDownload Function interface

Function outputs

Once the download is complete, *TAMS* provides a traffic file with the trajectories of every route found during the download session.

This file is created with the same format than the traffic files from EUROCONTROL, so the information of route trajectories can be analyzed or modified in the same way. It is also compatible with other modelling programs like SAAM.

6.4 General Overview

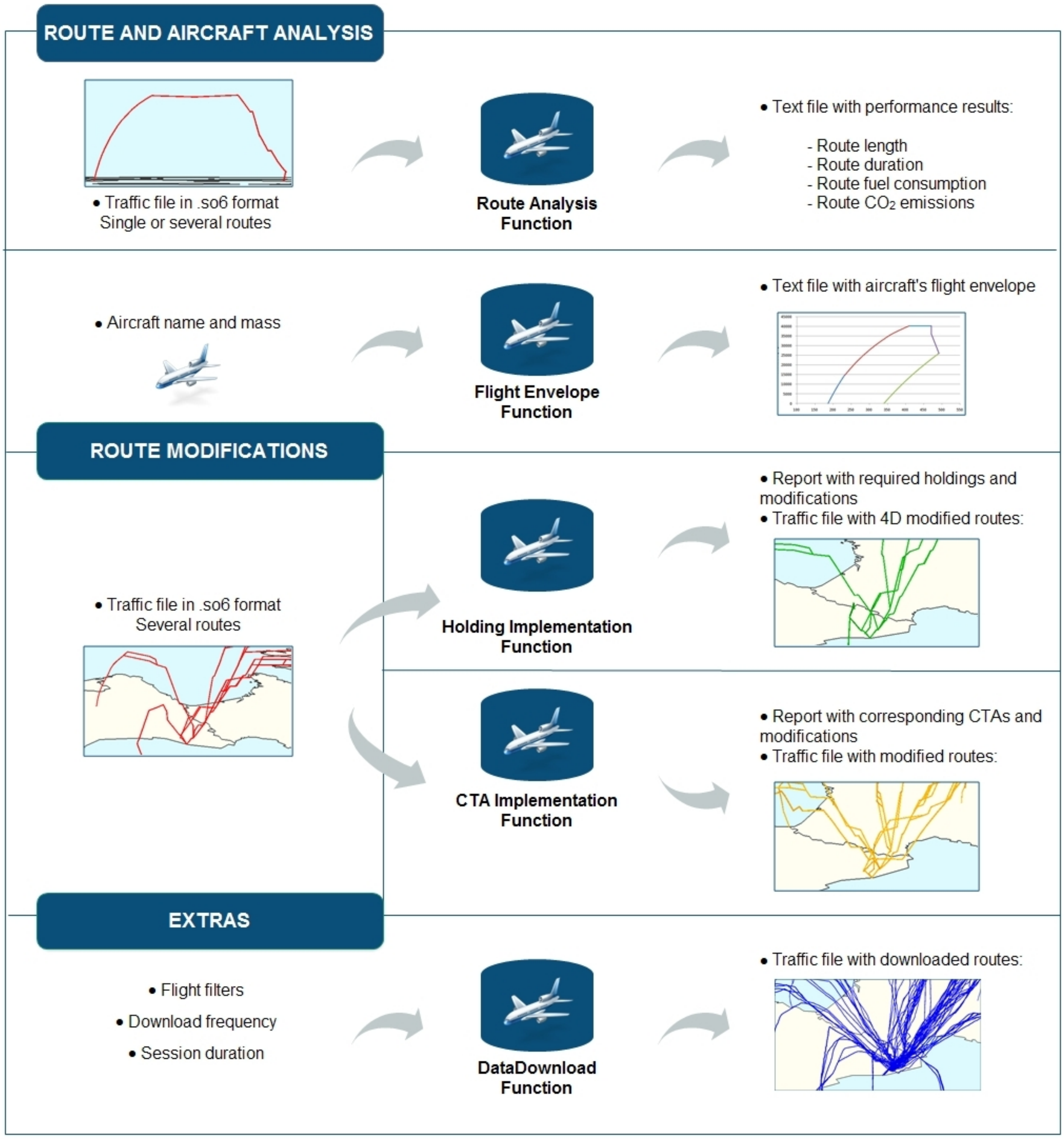


Figure 6.21: Overview of *TAMS*'s functions

Chapter 7

Software Validation

7.1 Validation Objectives

Any new software development needs to be submitted to validation phase in order to find possible mistakes. It is an essential procedure with the final objective to ensure its reliability.

The functionality that has been summited to a more exhaustive validation is the *Route Analysis* function, because of its relevance in *TAMS* as well as the possibility to validate it quantitatively. This chapter will focus on the validation of this function.

SAAM (see *Section 3.4: SAAM: System for traffic Assignment and Analysis at a Macroscopic level*) will be used as a reference for the validation of *TAMS*. The procedure will consist of analyzing several routes with both programs: *TAMS* (the program to be validated) and SAAM. These two programs have different procedures to analyze route trajectories, with distinct hypotheses and criteria. For this reason, the results obtained with *TAMS* cannot be expected to be identical that the ones provided by SAAM. The purpose of this comparison is to identify the causes of the discrepancies: if they are caused by the difference in the analysis procedure, the discrepancy will be justified. Otherwise the difference will be considered as an error of the software and it will have to be rectified.

7.2 Validation Run

7.2.1 Validation scenario

The input file in the validation process has to contain a number of representative routes in order to ensure an exhaustive validation. It also has to be similar to the ones that will be used in *TAMS* application. For these reasons, the traffic file summited to validation will be the next one:

- Number of flights: 288
- Destination of flights: LEBL Airport
- Date: 01.01.2014
- Type of traffic file: Scheduled (m1)
- File name: 20140101_m1_LEBL

7.2.2 Different analysis procedures

So far it has been commented the Step by Step method used by *TAMS* together with the BADA model. The combination of a method and a model define a technique for the analysis.

SAAM is also based on BADA but it uses a different method to analyze routes. This other method has been reproduced in *TAMS* to compare both results.

In order to justify the differences observed in route calculations, another analysis technique has been used. It is a combination of the two techniques: it uses SAAM's analysis method to calculate general results but instead of using SAAM's model (which is a combination between a simplified BADA model and other sources), it uses exclusively the simplified BADA model. The usefulness of this technique will be explained below.

For a clearer view of the three techniques, see Figure **??**. It reflects their corresponding methods and models.

7.2.3 Analysis for validation

Every route in the traffic file will be summited to four different types of analysis using the three techniques mentioned above. Those are:

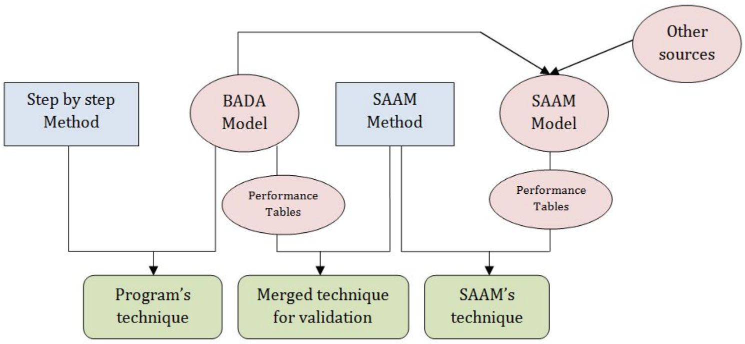


Figure 7.1: Chart relating each technique with its model and method.

- **Analysis 0.** With the original SAAM program.
    - Program: SAAM.
    - Analysis technique: SAAM's method with SAAM's model.
    - Objective: These results will be used as the reference values.
- **Analysis 1.** SAAM Copy: A code simulating SAAM's technique (both model and method) has been implemented in *TAMS*.
    - Program: *TAMS*.
    - Analysis technique: SAAM's method with SAAM's model.
    - Objective: If the results obtained with the original SAAM and the copy implemented in *TAMS* are exactly the same, it means that SAAM's technique has been correctly identified.
- **Analysis 2.** Merged technique:
    - Program: *TAMS*.
    - Analysis technique: SAAM's method with BADA model.
    - Objective: The differences between these results compared to the ones of Analysis 1 must be due to a discrepancy between SAAM's aircraft models and BADA's.
- **Analysis 3.** Program's default technique, without considering mass variations.
    - Program: *TAMS*.
    - Analysis technique: Step by step method and BADA model.
    - Objective: The differences between these results and Analysis 2 results must be due either to the simplification in the BADA model from Analysis 2 or to the differences between *Step by Step* method and SAAM's method.

7.2.4 Possible causes of discrepancies

The possible causes that may lead to significant differences between the analysis are explained below. When one of these causes occur in the analysis of a route, a function has been implemented to *TAMS* in order to show a notification. Those functions (or causes) are:

- **Missing aircraft information in one of the models**. It is possible that either SAAM or BADA model do not have performance information about a concrete aircraft. In this case, the comparison would not be possible.
- **BADA and SAAM Performance tables are totally unrelated**. It may be possible because SAAM's model is a combination of BADA model and other sources. For this reason, in some cases the performance of an aircraft in SAAM and BADA model may be unalike.
  This situation would justify the discrepancies between Analysis 1 and 2. A function called *comparePerformanceTables* has been implemented in *TAMS* to identify those aircrafts with significantly different performance tables.
- **Different aircraft assignation with BADA and SAAM**. Not all aircrafts have their own model, some of them are redirected to another aircraft (called synonym) with equivalent performance characteristics. However, this assignation may not be the same in BADA model and SAAM model.
  This situation would justify the discrepancies between Analysis 1 and 2. A function called *compareSynonyms* has been implemented in *TAMS* to identify those aircrafts with different model assignation.
- **Route with cruise segment below FL30**. In this case, SAAM method would ignore the consumption and emissions of those segments. This situation would cause discrepancies between Analysis 2 and 3. In the *Route Analysis* function, a counter has been added to indicate how many segments in the route are below FL30 in cruise configuration.
- **Null Thrust Descent Coefficients in BADA model**. Some aircrafts do not have these coefficients, which are used in thrust descent calculation.
  This situation would cause discrepancies between Analysis 2 and 3. In the *Route Analysis* function, a function has been added to indicate if the route aircraft have these coefficients nulls.
- **Partial Route Trajectory**. In case the input file proceeds from real trajectory data, it is possible that the trajectory is not entirely defined. In other words,

it only contains part of the route. In this case, SAAM's method would lead to significant errors in route calculation.
This situation would cause discrepancies between Analysis 2 and 3. In the *Route Analysis* function, a function has been added to indicate how many segments of the route belong to cruise, climb or descent configuration. These counters may indicate if the route is not completed.

7.3 Validation Criteria

The criteria established referring to error percentages between analysis is the following:

a) Between analysis 1 and 0 (SAAM copy and SAAM original), the maximum error allowed is of 1%. Since it pretends to be an exact reproduction of SAAM technique, only a small margin of discrepancy is accepted.

b) The percentage of error allowed in analysis 2 and 3 with respect to 0 is fo 10%. Since these analysis use different model and/or methods, the obtained results do not have to be identical. All other errors between 0 and 10% will be considered acceptable.

c) In case the differences are justified with one of the functions previously mentioned, they will not be considered as an error in the software.

7.4 Validation Results

The four analysis previously commented have been applied to the traffic file from January 1^{st} with its 288 flights, using SAAM for Analysis 0 (reference) and *TAMS* for Analysis 1 to 3. Theoretically, all the discrepancies between reference results must be justified with one of the causes explained before and the corresponding notification should be activated.

- *Analysis 1* (SAAM copy) conflictive flights : All flights reproduce identical results with respect to SAAM original results, no conflictive flights are found.
- *Analysis 2* (Merged Technique) conflictive flights :

Flight	**MaxError**	**Justified**	**Cause of discrepancy**
SIA378	12%	YES	Significantly different performance tables
LGL35Z	13,2%	YES	Significantly different performance tables
VQBGA	110,1%	YES	Different aircraft assignation

Table 7.1: Analysis 2: Conflictive flights

- *Analysis 3* (*TAMS* technique) conflictive flights. The first three flights are the same ones found in Analysis 2 comparison with an extra error, since the error is accumulated as we use a more different method from the previous one.

Flight	**MaxError**	**Justified**	**Cause of discrepancy**
SIA378	12%	YES	Significantly different performance tables
LGL35Z	34,2%	YES	Significantly different performance tables
VQBGA	111,6%	YES	Different aircraft assignation
AFR1348	16,3%	YES	Different aircraft assignation
VLG7333	13,2%	NO	
BEY131H	13,9%	YES	Thrust coefficients nulls
AFR1048	16,3%	YES	Different aircraft assignation

Continues on next page ↪

Flight	**MaxError**	**Justified**	**Cause of discrepancy**
AEA5454	38,1%	YES	Thrust coefficients nulls
YUBZZ	12,4%	YES	Thrust coefficients nulls

Table 7.2: Analysis 3: Conflictive flights

7.5 Validation Conclusion

The validation process performed in the previous section leads to a list of conclusions:

- SAAM analysis technique is correctly identified and represented in the *TAMS*, which means that the deduction of SAAM's technique together with its hypotheses and criteria are correct.
- 3,12% of flights in this traffic file differ between SAAM results and *TAMS* results, due to the discrepancies between both analysis techniques. Since the causes of these differences have been identified, they are not considered as an error of the software, just a different criteria applied in the calculations.
- The number of flights which has not been possible to identify the cause of the error represent a 0,3% of total flights. Since it is a really small percentage, it is not considered as relevant.
- As the analysis technique distances from the original SAAM technique, errors between results are accumulated and increased as it was expected.

With these conclusions, the validation process is considered complete.

Part III

Use Case: Software Application

Chapter 8

Use Case Definition

8.1 Purpose

This part corresponds to the application of the software developed to a use case of analysis. More concretely, the effects associated with the implementation of CTA have been studied. The airport of analysis has been Barcelona - El Prat (LEBL). This study has been performed with the collaboration of air traffic controllers from Gavà, who have offered their expert advices to select the appropiate hypotheses for the analysis.

The study has been performed with the comparison of three different scenarios:

First Baseline scenario, which corresponds to the study of the current situation of Air Traffic Management at LEBL.

Second Sequencing by holding scenario. It is a simulated scenario that pretends to represent the worst case scenario situation.

Third Sequencing by CTA scenario. It is a simulated scenario that pretends to represent the situation at LEBL airport with the implementation of the CTA.

The comparison between scenario 2 and 1 will be used to observe the efficiency of the LEBL airport nowadays. On the other side, the comparison between scenario 3 and 1 will be used to observe the improvements that could imply the use of sequencing by CTA.

The days of study with their selected time slots are shown in next chapter.

8.2 Scenario 1: Baseline

The analysis of this scenario follows the next characteristics:

- It is a non-simulated scenario
- This scenario reflects the current situation at LEBL airport: delay is absorbed with speed adjustments, vectorings and, only in high demand situations, holdings.
- In case of light air traffic, route trajectory reductions and Continuous Descent Operations are also applied by air traffic controllers.
- Input file: the input file for this scenario are real route trajectories downloaded from FlightRadar24 website. Traffic files provided by EUROCONTROL are not accurate enough to represent the holdings and vectorings of the current situation. Instead, FlightRadar24 website displays real time information about aircraft trajectories around the world with higher accuracy. For this reason, a functionality has been added to *TAMS*: the possibility to connect to this website and download the information of interest for the user in real-time and store it in a traffic file. Figure 8.1 shows the website information transformed into a traffic file by *TAMS*.
- The traffic file downloaded with *TAMS* in the selected time window has been analyzed with the Route Analysis function.

Figure 8.1: FlightRadar24 website (left) and traffic file downloaded by *TAMS* (right) corresponding to a single route

8.3 Scenario 2: Sequencing by holding

The second scenario follows the next characteristics:

- It is a simulated scenario
- It represents the worst case situation: delay is absorbed only with holdings, no other modifications are applied by air traffic controllers.
- Input file: m3 file of the day of study, filtering the flights corresponding to the selected time slot.
- Modifications: *TAMS* uses Holding Implementation function to introduce the required holdings. Hypotheses:
  - *TAMS* identifies the arrival time of each flight to TEBLA waypoint, which is the point where the queue will be applied.
  - *TAMS* establishes a separation time between each aircraft equivalent to a 5 NM separation (separation time will depend with the type of aircraft).
  - Each flight has associated a delay depending on the separation time that *TAMS* has computed for each flight. This delay will be absorbed in the corresponding Initial Approach Fixed (IAF), characteristic of each flight.
  - Holdings are performed at the IAF, starting with a Flight Level (FL) 70 altitude and leaving a vertical separation of 1000 ft between flights in case several holdings occur simultaneously at the same IAF.
  - Once the aircraft has absorbed its delay, it continues with the trajectory established in the original traffic file until landing.
- The traffic file generated by *TAMS* simulating this scenario is analyzed with the Route Analysis function.

8.4 Scenario 3: Sequencing by CTA

The last scenario is characterized by:

- It is a simulated scenario
- It represents the implementation of the CTA at LEBL airport.
- Input file: m3 file of the day of study, filtering the flights corresponding to the selected time slot.
- Modifications:*TAMS* uses CTA Implementation function to introduce CTA necessary modifications. Hypotheses:

  - The procedure followed by *TAMS* to associate a delay to each aircraft is the same that in scenario 2.
  - CTA constraint is the arrival time to TEBLA that *TAMS* has assigned to each aircraft.
  - To absorb delay, the aircraft reduces its speed with the necessary modifications to achieve TEBLA waypoint at the indicated time.
  - The distance where the aircraft are notified of their CTA is adjustable by the user. In the case of study this distance equals 100 NM.
  - The original trajectory of the aircraft (from the m3 file) is not modified. Route modifications only are applied in terms of speed.
- The traffic file generated by *TAMS* simulating this scenario is analyzed with the Route Analysis funciton.

8.5 Use Case Hypotheses

The study has been executed according to the following general hypotheses:

- Runway configuration: RWY 25R/25L, since it is the conventional configuration for arrivals at LEBL airport (see Figure 8.2).
- Runway maximum capacity: 40 aircrafts per hour.
- Routes have been analyzed only from TOD. Since *TAMS* modifies trajectories only nearby the airport, modifications in climb or descent phases are not relevant for this study and can contaminate the results.

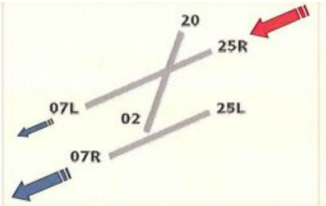


Figure 8.2: Runway configuration for the use case analysis

The characteristics of the selected time slots for the study are summarized in Table 8.1. All days have been selected randomly to study all possibilities.

Number	**Date**	**Day**	**Slot**	**Duration** [min]	**Flights**
1	04.05.2014	Sunday	19.00 - 21.00	120	30
2	05.05.2014	Monday	18.45 - 20.15	90	17
3	01.06.2014	Sunday	19.00 - 21.30	150	40
4	02.06.2014	Monday	19.00 - 21.00	120	27
5	12.06.2014	Thursday	17.00 - 21.00	240	68
6	13.07.2014	Sunday	19.00 - 22.00	180	68
7	20.07.2014	Sunday	16.10 - 20.50	280	74
8	30.07.2014	Wednesday	18.00 - 19.30	90	23
9	31.07.2014	Thursday	16.15 - 20.30	255	64
10	08.08.2014	Friday	16.15 - 21.10	295	90
			TOTAL	1820	501

Table 8.1: Selected days and time slots for the study

Chapter 9

Use Case Analysis and Results

This chapter starts with a detailed comparison between the scenarios studied. The analysis is focused on a single day to observe the particular behavior of each flight: Day 6 (13.07.2014), a 3 hour slot with 68 flights. It has been considered representative since its general results are similar to the total results of the study.

Finally, an overview of the general results is done at the end of the chapter. It is important to mention that all percentages and absolute values are referred to the *descent* phase of the flight, since routes have been analyzed from TOD.

9.1 Baseline Scenario

9.1.1 Demand

The number of aircraft inside Barcelona TMA during the day of analysis have been represented in Figure 9.1, in order to justify the results obtained in the following parts. As it is shown, highest demand corresponds to a 20-minute peak from 20:50 to 21:20 local time with 13 aircrafts inside the TMA flying to LEBL. Lowest demand corresponds to the end of the slot (22:00).

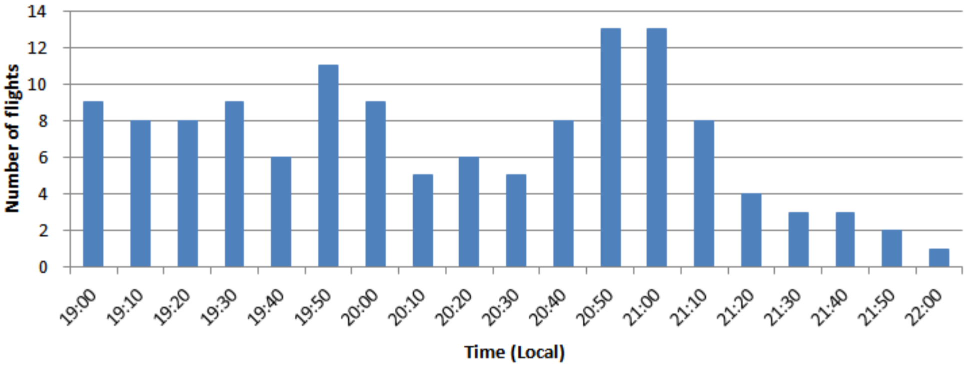


Figure 9.1: Number of aircrafts inside Barcelona TMA

9.1.2 Path stretching

Real trajectories from baseline scenario have been compared with their corresponding scheduled flight (m1) in terms of total length. Results for the day of analysis can be found in Table 9.1.

% Path stretching	**Flights**
[-40, -30)	1
[-30, -20)	1
[-20, -10)	6
[-10, 0)	4
[0, 10)	5
[10, 20)	19
[20, 30)	19
[30, 40]	13

Table 9.1: Percentage of path stretching in real flights compared to scheduled flights

Effectively, path stretching is commonly used: 56 flights (which corresponds to a 82 % of all slot flights) have expanded their trajectory and 13 flights have enlarged it more than a 30%. Flights with higher elongation and higher reduction respectively have been represented using SAAM to observe both scheduled and real trajectories and identify possible causes. Scheduled routes are represented in purple and real routes in green.

Figure 9.2 corresponds to a flight whose real route has been enlarged 31% with respect to scheduled. This case exemplifies the vectoring service provided by ATC to absorb delays. Figure 9.3 shows another flight with high enlargement. This case corresponds to a 30% stretching and it is due to holdings.

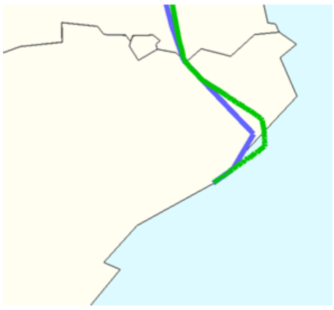


Figure 9.2: Flight BAW486 with 31% of path stretching

Figure 9.3: Flight EZY63DR with 30% of path extension due to holding

On the other hand, flight with most path reduction are shown in Figure 9.4 and Figure 9.5. The last one is caused by a continuous descent applied instead of a step down approach. Both modifications have been applied in lower demand timing.

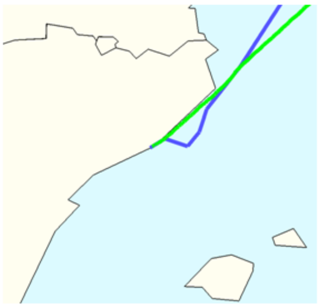


Figure 9.4: Flight AZA080 with 24% of path reduction

Figure 9.5: Flight EZY33XT with 35% of path reduction

9.1.3 Holdings

The number of holdings during the time slot has also been computed and it is shown in Figure 9.6. A total of 11 holdings have been observed in the 3 hour window.

Results exposed are consistent with the two demand peaks observed in Figure 9.1. This fact confirms the hypotheses of this first scenario: holdings are executed in high demand slots but not with high accumulations.

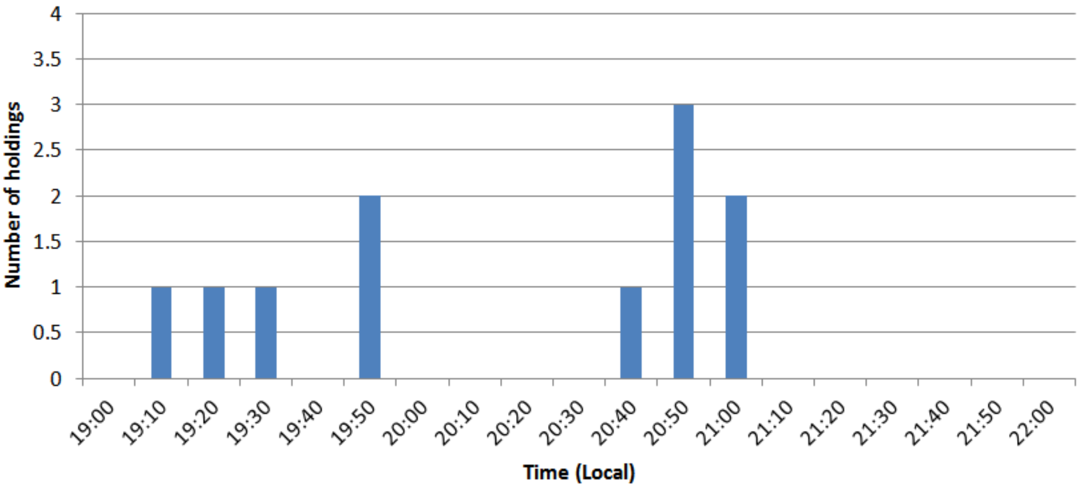


Figure 9.6: Number of holdings during time slot

9.2 Baseline vs Sequencing by holding Scenarios

The Sequencing by holding scenario is analyzed and compared to baseline to observe the benefits of the current situation in LEBL airport.

9.2.1 Delay and holdings

The first result presented is the delay assigned to each flight by *TAMS* to respect the minimum separations. Flights sorted by they arrival time with its corresponding delays are shown in Figure 9.7.

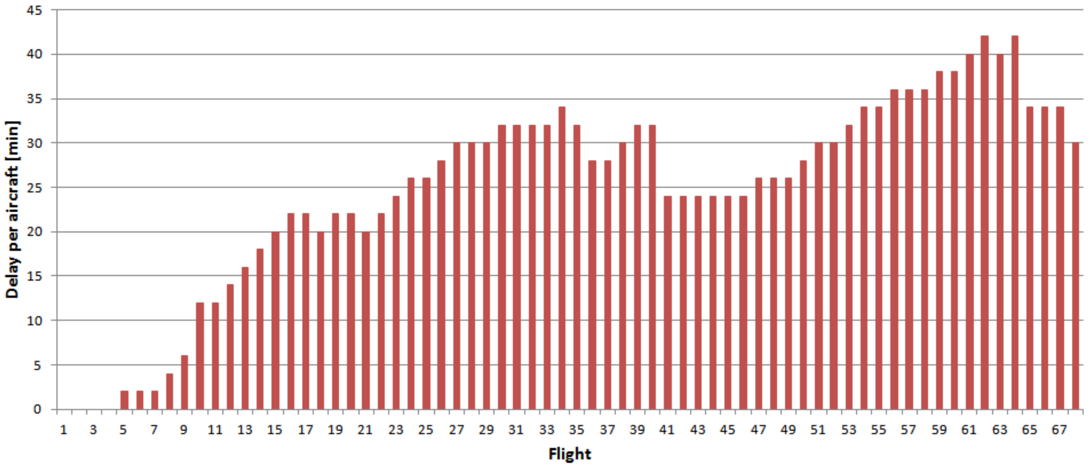


Figure 9.7: Delay per flight in Sequencing by holding scenario

It can be seen that delay increases with time, accumulating up to 42 minutes of holding. This high delay implies that simultaneous holdings must occur at their corresponding IAF, as it is shown in Figure 9.8. Comparing Figure 9.6 with this case, a significant increase of holdings can be observed: from 3 in the baseline scenario to 15 simultaneous in the sequencing by holding.

9.2.2 Fuel consumption

A comparison between baseline and sequencing by holding scenario is done in terms of fuel consumption. All flights in baseline scenario represent high reductions in terms of fuel consumption compared with this worst case scenario. The only exception are two flights where no delay is applied in holding scenario, so the comparison is done directly between original trajectories.

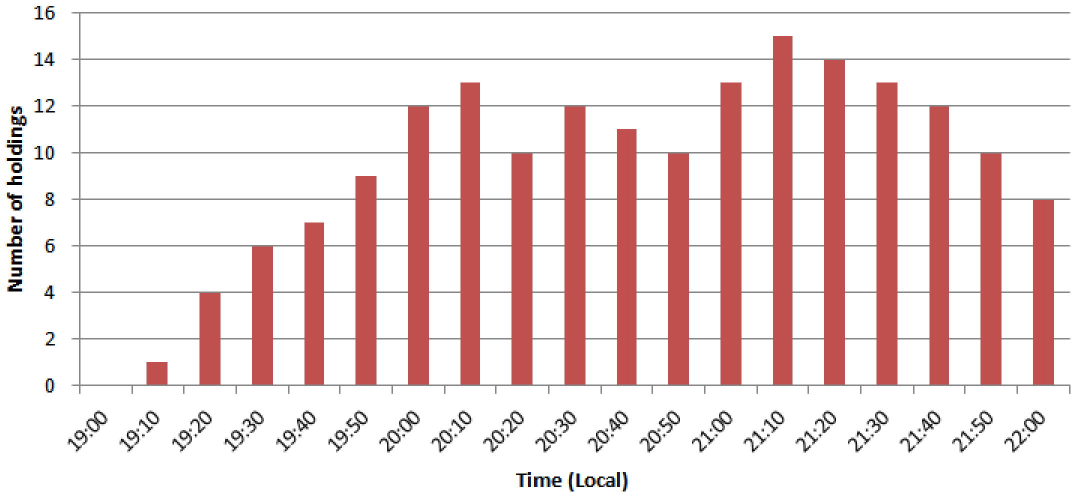


Figure 9.8: Number of holdings during time slot in baseline and holding scenarios

The flight with maximum savings in baseline corresponds to 89% of fuel savings. As expected, it is also the flight with maximum delay associated (42 minute delay).

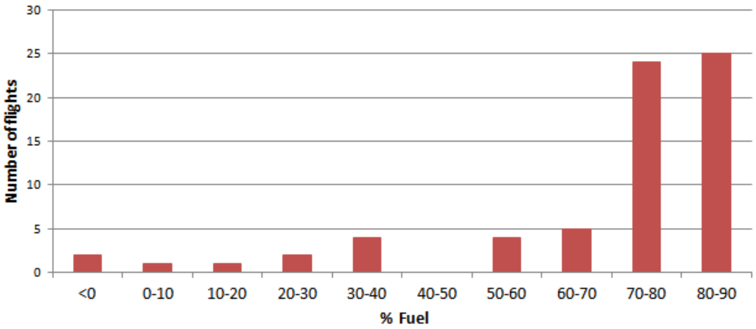


Figure 9.9: Baseline scenario fuel savings with respect to Sequencing by holding scenario

The general results in terms of fuel consumption for this first comparison in shown in Table 9.2:

Baseline scenario compared with Sequencing by holding	
Total routes analyzed	68
Time horizon	180 min
Fuel savings	63263 kg - 75,4%
Fuel savings per flight	930 kg/flight
Fuel savings per hour	21087 kg/hour

Table 9.2: General fuel consumption comparison of the day of analysis

9.3 Baseline vs Sequencing by CTA scenarios

The second analysis is done comparing the Baseline scenario with the Sequencing by CTA, to observe the benefits that CTA implementation could introduce.

9.3.1 Delay

TAMS has associated a CTA to each aircraft and its corresponding delay. These delays are significantly reduced compared to the delays of the Sequencing by holding scenario, as shown in Figure 9.10 (maximum delay in CTA is a 4 minutes delay, compared with the 42 minutes of the previous scenario).

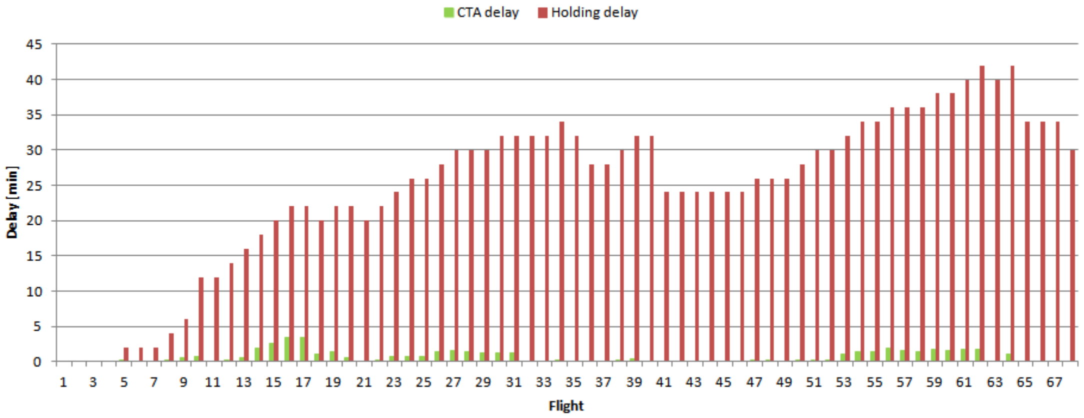


Figure 9.10: Comparison of delay per flight in Sequencing by CTA and by holding scenarios

The main difference between both flight sequencing is that in holding scenario delay accumulates with time. If an aircraft has associated a 15 seconds delay, following the scenario hypotheses it corresponds to a 2 minute holding, which increments the timing separation for the next aircraft in 1min 45sec that otherwise in CTA scenario would not be necessary. This fact makes Sequencing by CTA much more efficient in terms of delay.

9.3.2 Fuel consumption

The distribution of the savings in fuel consumption of CTA scenario is shown in Figure 9.11. Most flights (49/65) would experience positive savings with the CTA implementation, up to a 46%. Otherwise, the rest of flights (19/65) are not benefited from this sequencing concept, arriving to a 64% increment. This results are explained below.

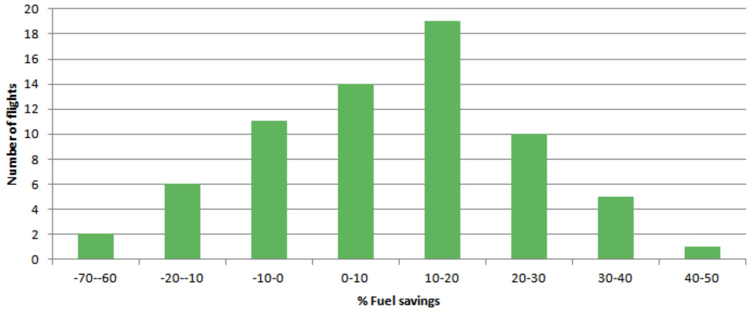


Figure 9.11: Sequencing by CTA scenario fuel savings with respect to Baseline scenario

The factors that affect fuel consumption are commented below.

1. Influence of step down approach

Flight VLG8021 is the one with higher losses with CTA in the slot analyzed. Its trajectory is represented in Figure 9.12. The main difference observed is that the trajectory from the simulation is following a step down approach while the real route is performing a more continuous descent. This fact increases consumption considerably since in cruise phase the fuel flow is higher than in descent. This difference can be observed in Figure 9.13, where the kilograms of fuel per each segment is compared. In the first 50 NM of the descent, the continuous descent has consumed 12 kg of fuel while the step down approach needed 134 kg.

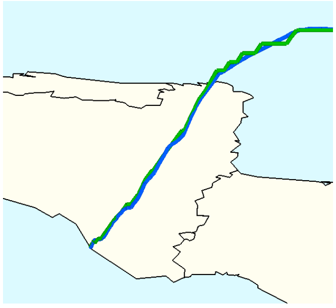


Figure 9.12: Flight VLG8021 descent trajectories: Scenario 3 (green) and Baseline (blue)

It is important to mention that the 3D trajectory of the route is not modified in CTA Implementation Function, so the consumption differences in this flight are not caused by the CTA Implementation.

2. Speed modification

Speed reduction in descent phase does not always imply less consumption. Figure 9.14

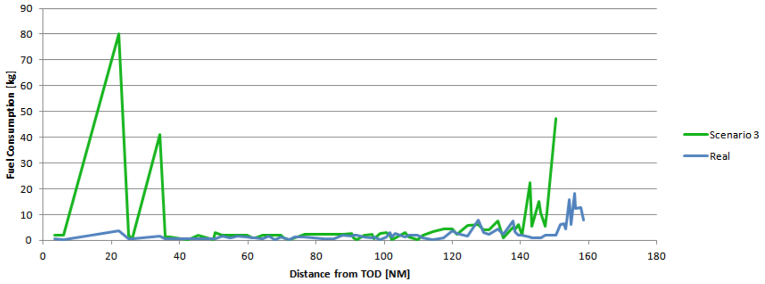


Figure 9.13: Consumed fuel comparison between both scenarios, flight VLG8021

shows the speed profiles of flight VLG8021 in both scenarios. It can be seen how the CTA Implementation Function has reduced considerably the aircraft speed from 100 NM before landing. Relating this figure with Figure 9.13, fuel consumption increases during the last part of the descent compared to the real route. This fact is caused by the fact that descending at lower speeds, the aircraft has to start approach and landing configurations (e.g. using slats and flaps, increase thrust) which increments drag and therefore fuel consumption.

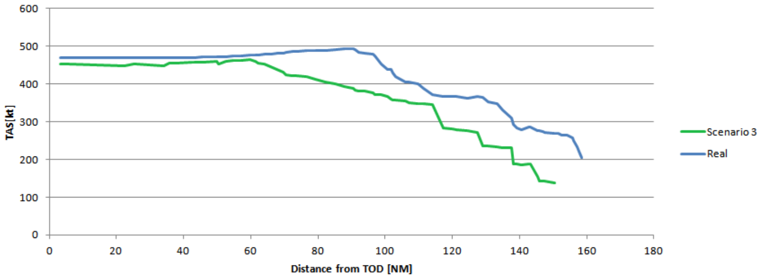


Figure 9.14: Speed profile comparison between both scenarios, flight VLG8021

3. Holding avoidance

The flights that get the most benefit of this CTA implementation are usually the ones with holdings in the baseline scenario. This implementation eliminates the use of holdings to absorb delay, since the flight arrives at the metering fix at the required time to continue with its route.

Figure 9.15 represents both trajectories of flight WZZ202, the one with most fuel savings in Sequencing by CTA scenario (46%).

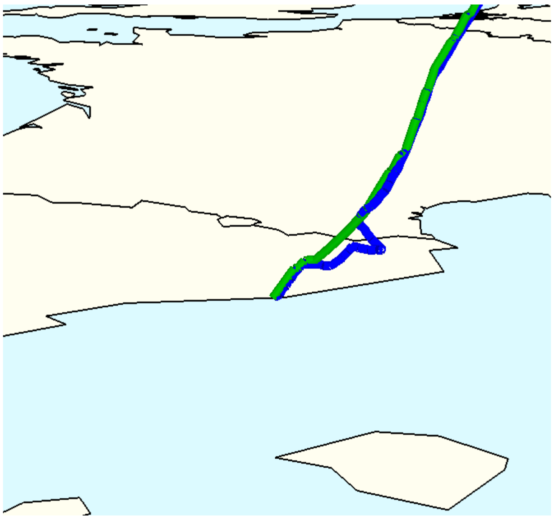


Figure 9.15: Flight WZZ202 descent trajectories

The combination of these three factors defines the savings or costs of the CTA scenario. For this day of analysis, the results in terms of fuel consumption are shown in Table 9.3.

Sequencing by CTA compared to Baseline scenario	
Total routes analyzed	68
Time window	180 min
Fuel savings	1784 kg - 8,6%
Fuel savings per flight	26,2 kg/flight
Fuel savings per hour	595 kg/hour

Table 9.3: General fuel consumption comparison of the day of analysis

9.3.3 Notification Distance

One last factor to analyze is the influence of the CTA notification distance. This factor affects two aspects: aircraft speed and fuel consumption.

Since the CTA assigned to each aircraft is independent of the notification distance, if the notification point (NP) is closer to the airport, the aircraft will be forced to reduce its speed abruptively than for further NPs. This fact can be observed in Figure 9.16, where two different notification distances have been compared: 70 NM and 120 NM. Also the original speed of the aircraft (without CTA implementation) and minimum speed are represented.

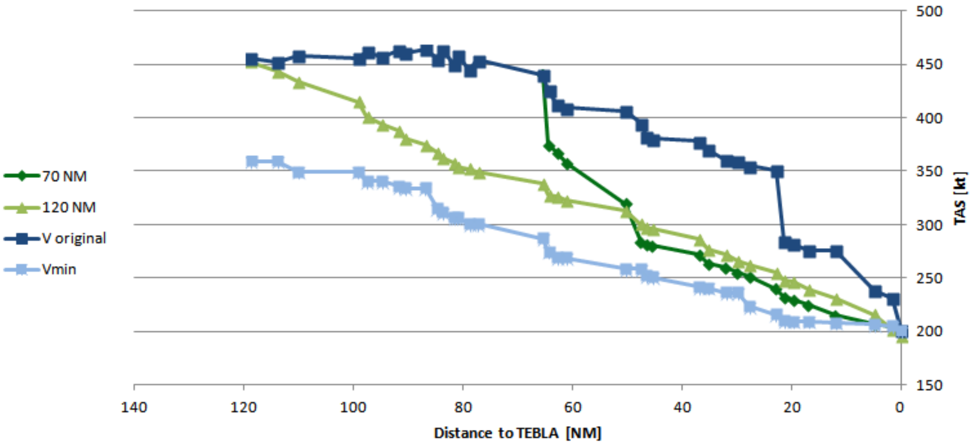


Figure 9.16: Speed schedule variation with NP distance

Several notification distances have been applied to this day of study (at the TMA entry, 100 NM, 125 NM, 150 NM and 200 NM) to observe how the total fuel consumption would be affected (see Figure 9.17).

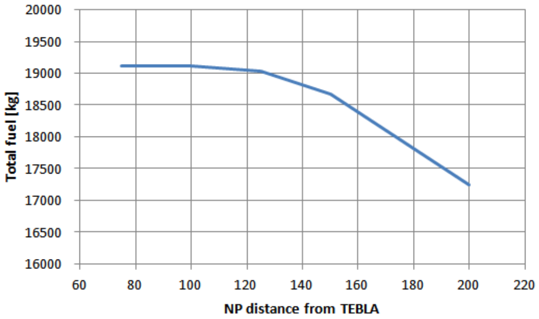


Figure 9.17: Total consumed fuel as a function of NP distance

9.4 General Results

In this section the general results of the study are summarized in terms of length, time, fuel and CO_2. Figure 9.4 shows the percentage results of savings between scenarios for each day of the study:

.					Savings Seq by CTA compared to Baseline			
Date	Length	Time	Fuel	CO_2	Length	Time	Fuel	CO_2
04.05.14	28.2%	37.0%	58.3%	58.3%	9.5%	7.5%	14.7%	14.7%
05.05.14	5.9%	14.8%	33.3%	33.3%	14.7%	12.1%	16.3%	16.3%
01.06.14	63.7%	67.7%	85.2%	85.2%	7.4%	10.4%	6.8%	6.8%
02.06.14	13.7%	22.9%	48.7%	48.7%	9.2%	6.2%	2.5%	2.5%
12.06.14	39.5%	47.7%	71.4%	71.4%	9.3%	6.3%	-0.2%	-0.2%
13.07.14	45.8%	52.9%	75.2%	75.2%	7.9%	8.7%	8.6%	8.6%
20.07.14	39.3%	41.6%	65.8%	65.8%	9.0%	17.6%	20.4%	20.4%
30.07.14	38.0%	45.5%	69.8%	69.8%	9.6%	11.2%	10.1%	10.1%
31.07.14	33.7%	40.8%	63.7%	63.7%	8.2%	9.1%	15.2%	15.2%
08.08.14	36.6%	43.3%	67.5%	67.5%	7.2 %	8.6 %	7.8%	7.8%

Table 9.4: Results of each scenario and day

Saving percentage of Baseline scenario compared to Sequencing by Holdings:

% Length	**% Time**	**% Fuel**	**%** CO_2
40.0	46.5	69.9	69.9

Table 9.5: Baseline scenario fuel savings for this study

Saving percentages of Sequencing by CTA compared to Baseline scenario:

% Length	**% Time**	**% Fuel**	**%** CO_2
8.6	10.0	10.5	10.5

Table 9.6: Sequencing by CTA scenario fuel savings for this study

Focusing into fuel consumption and CTA scenario, it is also possible to obtain the savings in kilograms per flight and per hour:

Total routes analyzed	501
Slot	1820 min
Fuel savings	16185 kg - 10,5%
Fuel savings per flight	32.3 kg/flight
Fuel savings per hour	534 kg/hour

Table 9.7: Mean fuel consumption savings with Sequencing by CTA scenario

Table 9.8 shows the percentage and minutes of use of both East and West configurations at LEBL airport, from January to July 2014.

Config	JAN	FEB	MAR	APR	MAY	JUN	JUL
West (%)	94.65	94	79.23	72.09	85.72	76.19	72.82
West (min)	28169	25266	23578	20762	25509	21943	21670
East (%)	5.35	6	20.77	27.91	14.28	23.81	27.18
East (min)	1591	1614	6182	8038	4251	6857	8090

Table 9.8: Percentages and minutes of use of West and East configurations at LEBL airport

Since East configuration has not been analyzed in this study, the results can only be applied to West configuration. Assuming that the same behavior as studies in the Use case replicates regularly through the whole semester, it would lead to:

Minimum savings per year: 2544328 kg of fuel

9.5 Conclusions

The conclusions of these results can be summarized in:

- The hypotheses for the Baseline scenario were consistent with the results: most flights enlarge their trajectory to absorb delay (82% in the day of the example), which reduces the necessity of holdings (11 holdings in the three hour slot).
- This situation is highly efficient compared with the worst case scenario (Sequencing by holdings). Baseline scenario reduces the maximum delay in the example day from 42 minutes to 11. This leads to a significant reduction in fuel consumption, with a 69.9 % mean value in the 10 days of the study. All analyzed days respond positively to the Baseline scenario.
- The implementation of CTA at LEBL airport could lead to a even more efficient traffic management. According to the studied use case, this could be of the order of 10%. Several factors are involved in the fuel consumption calculation, such as the notification point, the type of approach and the speed reduction. The combination of all them does not always imply fuel savings with the implementation of CTA, but the mean values show a positive response.

Part IV

Miscellaneous

Chapter 10

Project Planning

10.1 Project tasks list and description

1. Project planning
   - 1A Project phases definition
   - 1B Gantt diagram elaboration
2. State of the art and documentation
   - 2A BADA v3 capabilities: identification of BADA possibilities and specifications.
   - 2B SAAM capabilities: identification of SAAM functions and possible applications in the software.
3. Project requirements
   - 3A Program capabilities: definition of the required functions
   - 3B Calculation methods: theoretical calculations previous to code programming
   - 3C Programming language and IDE selection: analysis of different possibilities and selection of the most suitable for the project.
4. Tool development
   - 4A Functions description: inputs and outputs of each function.
   - 4B Code development: code writing of the different functions
   - 4C Validation: checking of correct structure and execution of each function.

5. Use case implementation

    5A Use case definition: hypotheses and characteristics of the case study.

    5B Development of required tools: in case use case requires extra functionalities of the software.

    5C Use case analysis: analysis of the results obtained and conclusions.

6. Project drafting: writing of the corresponding documents to deliver with the project.

10.2 Gantt Diagram

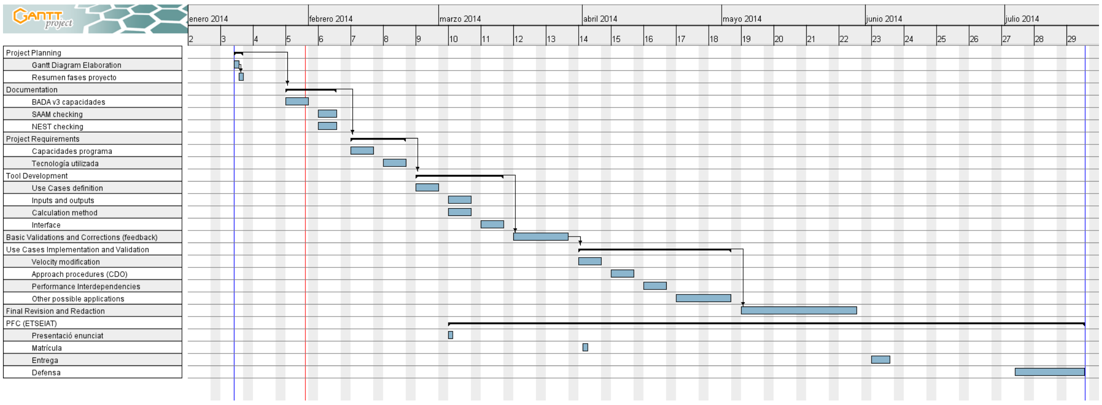


Figure 10.1: Gantt Diagram of the project

Chapter 11

About Advanced Logistics Group

This project has been entirely developed in Advanced Logistics Group (ALG), a transportation, infrastructure and logistics consultancy which offers strategic and integral consultancy solutions.

ALG has a strong international presence, with 2000 delivered projects in more than 50 countries worldwide in the last 25 years.

Referring to the aviation sector, ALG offers consulting services such as operations, infrastructure planning and development, or IT & technology systems. It has an extensive experience in aviation, which represents a 40% of the activity. The company has worked for airlines and general aviation operators, airports, ANSPs and public administrations. Its aviation team is composed by 40 aviation consultants.

In the field of Air Navigation Services, ALG has a large experience as strategic advisors with clients including AENA, NATS or EUROCONTROL.

Chapter 12

Possible Further Developments

TAMS is the first design of a performance analysis and modelling tool that can be easily extended to further applications. The combination of a potent programming language like C++, a model to simulate aircraft performance and a wide database of almost all aircrafts flying in the European airspace, lead to a powerful tool opened to a wide range of applications.

Referring to the functionalities that are already implemented, some improvements can be achieved. The speed profile assigned in the Sequence by CTA scenario can be modified to increase the fuel consumption savings, using a speed profile more similar to the aircraft optimum speed.

It could be also studied to extend the optimization to all phases of the flight, either in terms of optimum speed or optimum altitude to encounter the minimum fuel consumption.

The study of CTA implementation can be applied equally to other procedures in phase of validation and analysis. Some examples could be Continuous Descent Operations (CDOs) or Continuous Climb Operations (CDAs).

Finally, the implementation of *TAMS* with another automated tool developed within the same company could be studied. This other tool, *ProCAD*, is used for the design of navigation procedures at the airport that could be validated using *TAMS*.

Chapter 13

Conclusions

Both objectives of the project have been successfully achieved. On the one hand, *TAMS* has accomplished the requirements established at the initial phase. All functionalities have been implemented correctly, performing successfully the validation exercise.

During the development of the project new obstacles have appeared, which have evolved to new functionalities of the program and their corresponding added value (i.e. the DataDownload function).

TAMS has proved to be a reliable software for performance analysis, adaptable to user requirements and compatible with other existing simulating tools.

On the other hand, the application of *TAMS* in the use case has lead to interesting results. Barcelona - El Prat has been identified as an airport with efficient air traffic management, reflected by the corresponding savings.

Finally, the study of the CTA implementation effects at LEBL airport has shown a positive response. As a preliminary study, it supports the implementation of new ATM procedures that will lead to a modernization of Air Traffic Management in the European context.

The combination of a powerful programming language such as C++ with a widespread database like BADA has resulted in a flexible and reliable software tool. *TAMS* can be easily extended to the study of other new operational concepts, and be supportive to the R&D improvements to ATM operations of the European ATM Master Plan.

Bibliography

[1] Eurocontrol. `https://www.eurocontrol.int/`, September 2014.

[2] Sesar joint undertaking. `http://www.sesarju.eu/`, September 2014.

[3] The european atm master plan. `https://www.atmmasterplan.eu/`, September 2014.

[4] Laurence H. Mutuel. Initial 4d trajectory management concept evaluation. `http://www.atmseminar.org/seminarContent/seminar10/papers/249-Mutuel_0126130707-Final-Paper-4-12-13.pdf`, September 2014.

[5] EUROCONTROL. *Coverage of 2012 European air traffic for the Base of Aircraft Data - Revision 3.11*, August 2013.

[6] Introduction to programming. `http://cplus.about.com/od/introductiontoprogramming/p/programming.htm`, September 2014.

[7] List of programming languages. `http://en.wikipedia.org/wiki/List_of_programming_languages`, September 2014.

[8] Bruce Eckel. *Thinking in C++, Volume 1.* Prentice Hall, second edition, 2000.

[9] Comparison of programming languages. `http://www.cprogramming.com/langs.html`, September 2014.

[10] John Walkenbach. *Excel 2010 Power Programming with VBA.* Wiley Publishing, Inc.

[11] Qt Project. `http://qt-project.org/`, September 2014.

[12] Gui programming definition. `http://dictionary.reference.com/browse/GUI`, September 2014.

[13] Julian Templeman. *Microsoft Visual C++ Step by Step.* O'Reilly Media, Inc., 2013.

[14] EUROCONTROL. Demand Data Repository (DDR) Project. `http://www.eurocontrol.int/ddr`, September 2014.

[15] Aircrafts limits. `http://aerostudents.com/files/aircraftPerformance/aircraftLimits.pdf`, September 2014.