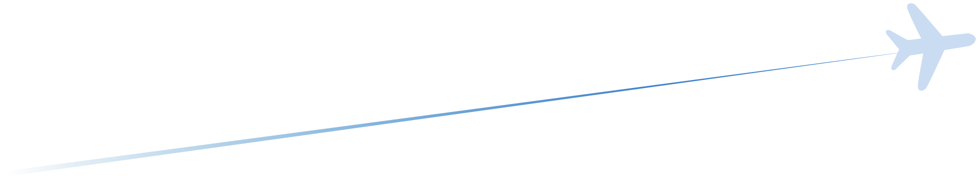


Report on the availability of the APACHE Framework

Deliverable D4.1

APACHE

Grant:	**699338**
Call:	**H2020-SESAR-2015-1**
Topic:	**ER-11-2015 (ATM performance)**
Consortium coordinator:	**Universitat Politècnica de Catalunya (UPC)**
Edition date:	**07th May 2018**
Edition:	**00.01.00**

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Xavier Prats – UPC	APACHE coordinator. WP4 leader	03/05/2018
Ramon Dalmau – UPC	UPC Researcher	19/01/2018
Cristina Barrado – UPC	UPC Researcher	19/01/2018
Yan Xu – UPC	UPC Researcher	19/01/2018
Fedja Netjasov – UB-FTTE	UB-FTTE Researcher	19/01/2018
Goran Pavlovic – UB-FTTE	UB-FTTE Researcher	02/05/2018
Andrija Vidosavljevic – ENAC	ENAC Researcher	02/05/2018

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Ignacio Agüi – ALG	Internal (non-contributor) reviewer.	07/05/2018
Andrija Vidosavljevic – ENAC	Internal reviewer.	07/05/2018
Fedja Netjasov – UB-FTTE	Internal reviewer.	04/05/2018
Xavier Prats – UPC	APACHE coordinator. WP4 leader	07/05/2018

Approved for submission to the SJU By — Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
Xavier Prats – UPC	APACHE coordinator. WP4 leader	07/05/2018

Rejected By – Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date

Document History

Edition	Date	Status	Author	Justification
00.00.01	07/11/2017	Draft	UPC	Initial structure and drafted sections.
00.00.02	19/01/2018	Draft	UPC, UB-FTTE	UB-FTTE contribution on the RA. UPC contribution on the TP and TCP.
00.00.03	02/05/2018	Draft	UPC, ENAC	ENAC contribution on the ASP. UPC/UB-FTTE contribution on the PA.
00.00.04	03/05/2018	Draft	UPC	Addition of Appendix E (requirements compliance) and generation of final Draft.
00.00.05	07/05/2018	Draft	ALG, UB-FTTE, ENAC	Review from internal contributing and non-contributing reviewers.
00.01.00	07/05/2018	Final	UPC	First release submitted to the SJU.

Dissemination level

This document is PUBLIC.

APACHE

ASSESSMENT OF PERFORMANCE IN CURRENT ATM OPERATIONS AND OF NEW CONCEPTS OF OPERATIONS FOR ITS HOLISTIC ENHANCEMENT

This Document[1] is part of a project that has received funding from the SESAR Joint Undertaking under grant agreement No 699338 under European Union's Horizon 2020 research and innovation programme.

Abstract

The APACHE project proposes a new framework to assess European ATM (air traffic management) performance based on simulation, optimization and performance assessment tools that will be able to capture the complex interdependencies between KPAs at different modelling scales. In this context, a new platform (the APACHE Framework) has been developed in the Project, which is the result of the integration (and enhancement) of different existing tools previously developed by some of the APACHE consortium members. This deliverable is the software availability note of the APACHE Framework.

This document firstly describes how the different system components have been integrated into a single workflow, aiming at fulfilling the requirements of the Project. Verification and integration tests of the whole APACHE Framework are presented, showing the successful integration of the different components. Then, validation tests of the individual components of the APACHE Framework are described, taking into account that the validation at system level (i.e. the validation of the whole APACHE Framework as a unified tool to assess ATM performance) is out of the scope of this Deliverable and will be reported in D5.1.

Supported by all these tests, the evaluation of the requirements identified in previous Deliverable D3.2 is presented, showing the evidences that proof the fulfilment of requirements and giving a rationale for those (very few) requirements not fulfilled or changed. Finally, this report concludes with a summary of all limitations and assumptions taken when developing the APACHE Framework, aiming at clearly identifying the maturity level of the developed Framework and pointing towards future enhancements and developments of the tool.

---

[1] The opinions expressed herein reflect the author's view only. Under no circumstances shall the SESAR Joint Undertaking be responsible for any use that may be made of the information contained herein.

Table of Contents

List of figures

List of tables

1 Introduction

1.1 Purpose, context and scope of the document

The APACHE Project covers the topic *ER-11-2015 – ATM Performance within the area of ATM Operations, Architecture, Performance and Validation* and proposes a new approach based on simulation, optimization and performance assessment tools, which aims to better capture performance in air traffic management (ATM), as well as the complex interdependencies and eventual trade-offs among different key Performance Areas (KPA).

This Deliverable *D4.1 – Report on the availability of the APACHE framework,* reports on the main outcomes of work package (WP) 4: *WP4 – Development of the APACHE framework*: it summarises the implementation done for this first release of the APACHE System and it details the integration, verification and component validation tests. Finally, it also summarises the list of assumptions and limitations of this particular implementation of the APACHE System, aiming at properly setting the scope of the Project validation exercises in *WP5 – Simulation and assessment*, but also to clearly identify gaps and room for improvements for eventual future developments of the APACHE Framework.

As it is shown in Figure 1-1, this document takes as main input the previous deliverable *D3.2 – Functional requirements and specifications for the ATM performance assessment framework* (APACHE Consortium, 2018), and serves as principal input of *WP5 – Simulation and assessment*, where the Project validation exercises will be performed, analysing a wide set of scenarios and case studies.

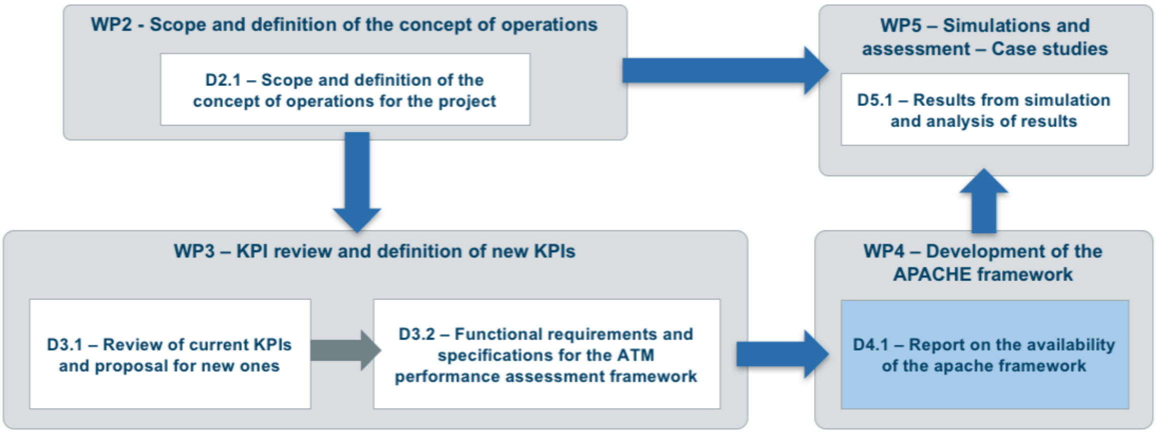


Figure 1-1. Context of deliverable D4.1

1.2 The APACHE Framework

As reported in the APACHE Project Deliverable D3.2 (APACHE Consortium, 2018), The APACHE project revolves around a novel framework that is expected to generate optimal trajectories, considering the business models of the airspace users; optimal airspace configurations, considering ANSP needs and constraints; and integrate both of them into an advanced air traffic flow management (ATFM) scheme. The enabling System can be configured to reproduce different modes of operation, representative of current ATM, or simulating (with certain limitations) the influence of future operational concepts.

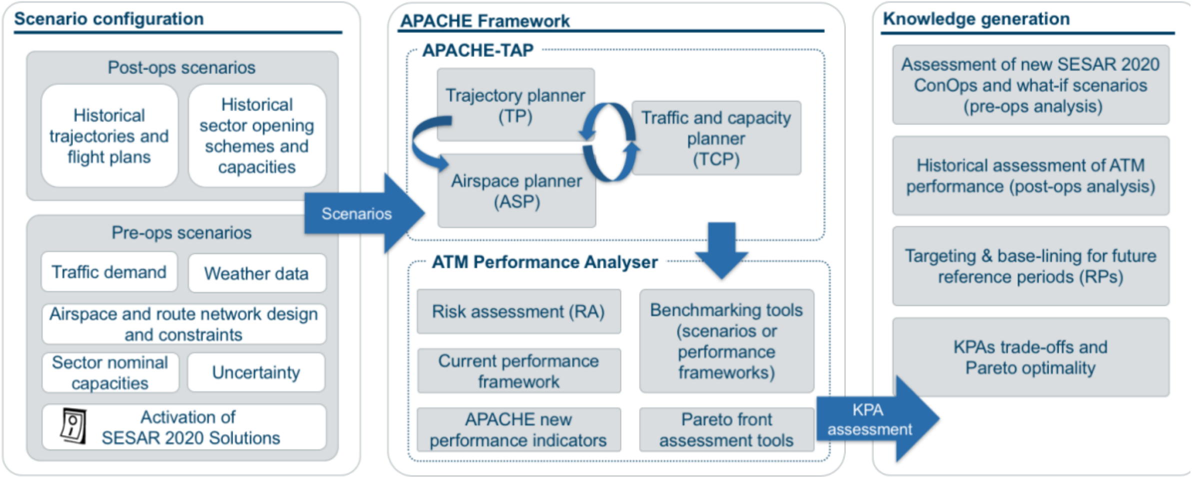


Figure 1-2. Context of the APACHE Framework within the APACHE Project

Figure 1-2 shows the overall concept of the whole APACHE Framework. First, several scenarios to be studied are defined, setting up different options regarding the demand of traffic, airspace capacities and eventual restrictions; SESAR solutions or future operational concepts to be simulated; and the level of uncertainty to be considered.

As detailed in (APACHE Consortium, 2018) two types of performance assessment are foreseen in this Project: **"Post-ops"** (monitoring) analysis, using scenarios created from historical data; and **"Pre-ops"** (planning) analysis, over synthesised scenarios with the purpose to enable "what-if" studies or the assessment of different ATM performance trade-offs.

As seen in Figure 1-2, the APACHE Framework consists of the integration of different software components. On one hand, the **Performance Analyser (PA)** module, which implements all the performance indicators (PIs) proposed in the APACHE performance framework, including as well some indicators from the current performance scheme for benchmarking purposes. On the other hand, the **APACHE-TAP (trajectory and airspace planner)**, which could be seen as a small prototype of an ATM simulator and having a double functionality in this Project:

- To support the implementation of novel ATM PIs, which require from some advanced functionalities (such as optimal fuel trajectories considering real weather conditions, optimal airspace opening schemes, large-scale conflict detection, etc.).
- To synthesize traffic and airspace scenarios representative enough of current operations; or emulating future operational concepts in line with the SESAR 2020 ConOps (i.e. one or more SESAR solutions enabled).

This double functionality of the APACHE-TAP is also shown in the block diagram of Figure 1-3.

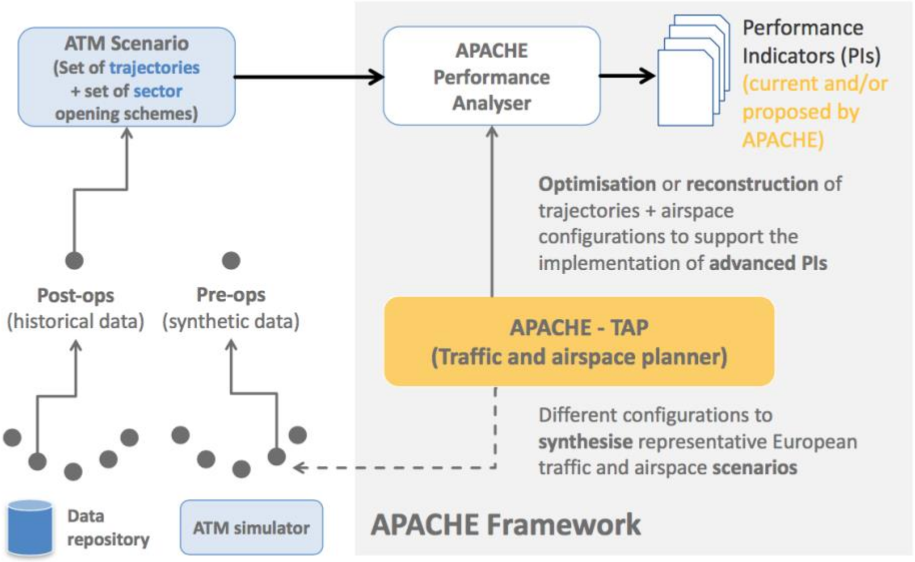


Figure 1-3. Double usage of the APACHE-TAP within the APACHE Framework

1.3 Document structure

The document is structured as follows:

- **Section 1:** Introductory section that outlines the context and purpose of this deliverable, containing also a glossary of terms.
- **Section 2:** This section summarises the integration and verification tests of the first release of the APACHE Framework. It includes the details and testing of the integration of the different components that form the whole workflow, as well as the details on the different interfaces between System components.
- **Section 3:** Devoted to show the validation exercises done at individual level, for each APACHE System component. It should be noted that the APACHE Validation exercises (i.e. validating the whole integrated Framework) are subject of WP5 and will be reported in D5.1.
- **Section 4:** Summarising the limitations and assumptions of the APACHE Framework.

1.4 Glossary

Term	Explanation
ACAS	Airborne Collision Avoidance System
ACC	Area Control Centre
AD	Arrival Delay
ADCB	Advanced Demand and Capacity Balance
AEQ	Airspace Equity Indicators
AH	Air Holding

Term	Explanation
AIP	Aeronautical Information Publication
AIRAC	Aeronautical Information Regulation and Control
AIS	Aeronautical Information Service
ANSP	Air Navigation Service Provider
ASM	Available Seat Mile
ASP	Airspace Planner (APACHE system component)
ATC	Air Traffic Control
ATCO	Air Traffic Controller
ATFCM	Air Traffic Flow and Capacity Management
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
ATS	Air Traffic Services
AU	Airspace User
BADA	Base of Aircraft Data
BRTE	Boeing Research and Technology Europe
CAB	Controlled Airspace Block
CAP	Capacity Indicators
CASA	Computer Assisted Slot Allocation
CAUTRA	French national aeronautical data repository
CCC	Continuous Cruise Climb
CE	Cost Efficiency Indicators
CFMU	Central Flow Management Unit
CI	Cost Index
CMB	Climb phase
CONF	Sector configuration
ConOps	Concept of Operations
CPA	Closes Points of Approach
CPR	Correlated Position Report
CPU	Central Processing Unit
CRZ	Cruise phase
CS	Collapsed Sector
CSV	Comma Separated Value
CTA	Controlled Time of Arrival
DAC	Dynamic Airspace Configuration
DCB	Demand and Capacity Balance

Term	Explanation
DDR2	Demand Data Repository
DES	Descent phase
DR	Delay Recovery
DYNAMO	DYNAMic Optimiser
EAD	Eurocontrol's European AIS Database
ECAC	European Civil Aviation Conference
ECMWF	European Centre for Medium-Range Weather Forecasts
ENAC	Ecole Nationale de l'Aviation Civile
ENV	Environmental Indicators
ER	Exploratory Research
ERBT	Executed RBT
ES	Elementary Sector
ETA	Estimated Time of Arrival
ETD	Estimated Time of Departure
FAB	Functional Airspace Block
FABEC	Functional Airspace Block Europe Central
FL	Flight Level
FLEX	Flexibility Indicators
FLIP	PEP's Flight Planning Module
FMP	Flight Management Position
FMS	Flight Management System
FR	Free Route
FRA	Free Route Areas
FUA	Flexible Use of Airspace
GAMS	General Algebraic Modeling System
GFS	Global Forecast System
GH	Ground Holding
GRIB	GRIdded Binary
HPC	High Performance Computing
ICAO	International Civil Aviation Organization
ICO	Improved Configuration Optimizer
IFR	Instrumental Flight Rules
IQR	Inter-Quartile Range
ISA	International Standard Atmosphere
KEA	Key performance Environment indicator based on Actual trajectory

Term	Explanation
KEP	Key performance Environment indicator based on last filed flight Plan
KPA	Key Performance Area
KPI	Key Performance Indicator
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LF	French Airspace
LH	Linear Holding
LI	Loop iteration (TP-TCP)
LRC	Long Range Cruise
LVL	Level
MLM	Maximum Landing Mass
MPI	Message Passing Interface
MTOM	Maximum Take-Off Mass
NCEP	National Centers for Environmental Prediction
NEST	Network Strategic Tool
NFS	Network File System
NMAC	Near Mid-Air Collision
NOAA	National Oceanic and Atmospheric Administration
O/D	Origin/Destination
OS	Operative System
PA	Performance Analyser
PAR	Participation Indicators
PC	Personal Computer
PCA	Principal Component Analysis
PEP	Airbus Performance Engineering Program
PF	Pareto Front
PI	Performance Indicator
QX	Quartile X
P2P	Peer to Peer
PRU	Performance Review Unit
RA	Risk Assessment (APACHE system component)
RAM	Random Access Memory
RBT	Reference Business Trajectory
RPAS	Remotely Piloted Aircraft Systems
SAAM	System for Airspace Analysis at Macroscopic level

Term	Explanation
SAF	Safety Indicators
SAM	Sharable Airspace Module
SBT	Shared Business Trajectory
SESAR	Single European Sky ATM Research
SJU	SESAR Joint Undertaking
SSH	Secure Shell
TAP	Trajectory and airspace planner module (main component of the APACHE system)
TBO	Trajectory Based Operations
TC	Test Case
TCAS	Traffic Collision Avoidance System
TCP	Traffic and Capacity Planner (APACHE system component)
TMA	Terminal Manoeuvring Area
TP	Trajectory Planner (APACHE system component)
UAS	Unmanned Aircraft Systems
UB-FTTE	University of Belgrade-Faculty of Transport and Traffic Engineering
UPC	Technical University of Catalonia (Universitat Politècnica de Catalunya)
VFR	Visual Flight Rules
WP	Work Package

Table 1-1. Glossary

2 APACHE System integration

As seen in Figure 1-2, the **APACHE System** consists of the integration of the following software modules:

- The **APACHE-TAP**, composed, in turn by:
  - the trajectory planner (TP) component, developed by UPC;
  - the airspace planner (ASP) component, developed by ENAC; and
  - the traffic and capacity planner (TCP) component, developed by UPC.
- The **Performance analyser (PA)** composed by:
  - the risk assessment component (RA) developed by UB-FTTE; and
  - a software suite, jointly developed by UB-FTTE and UPC, with various tools to compute Performance Indicators and enabling benchmarking, assessment and visualisation capabilities.

The three APACHE-TAP components and the RA component are enhanced versions of preliminary software tools or prototypes developed and owned by the corresponding Partner of the APACHE consortium. The PA component has been implemented from scratch within the APACHE Project.

As a consequence, these software components use different platforms, are coded in different programming languages and, in order to preserve the Intellectual Property Rights of each Partner, are stored and executed in different premises. This makes the APACHE System a highly heterogeneous and distributed prototype with a basic level of integration, which uses **plain text files as principal method to interchange information among the different components**. This level of integration was deemed appropriate and enough given the scope, purpose and maturity level of the APACHE Project.

This section describes how this integration has been done.

2.1 APACHE Framework workflow

Figure 2-1 shows a block diagram depicting the APACHE System integration, where the components workflow is shown, along with the different files serving as interface among components. Appendix A of this Document details and summarises all input/output interface files of the APACHE Framework workflow.

2.1.1 Start of the workflow: Trajectory Planner (TP)

The APACHE TP is composed by two main software components: The DYNAMIC Optimiser (**DYNAMO**) and the **Meta** Launcher (Meta for the remainder of the document). Appendix A.1 details the format of the TP input/output files.

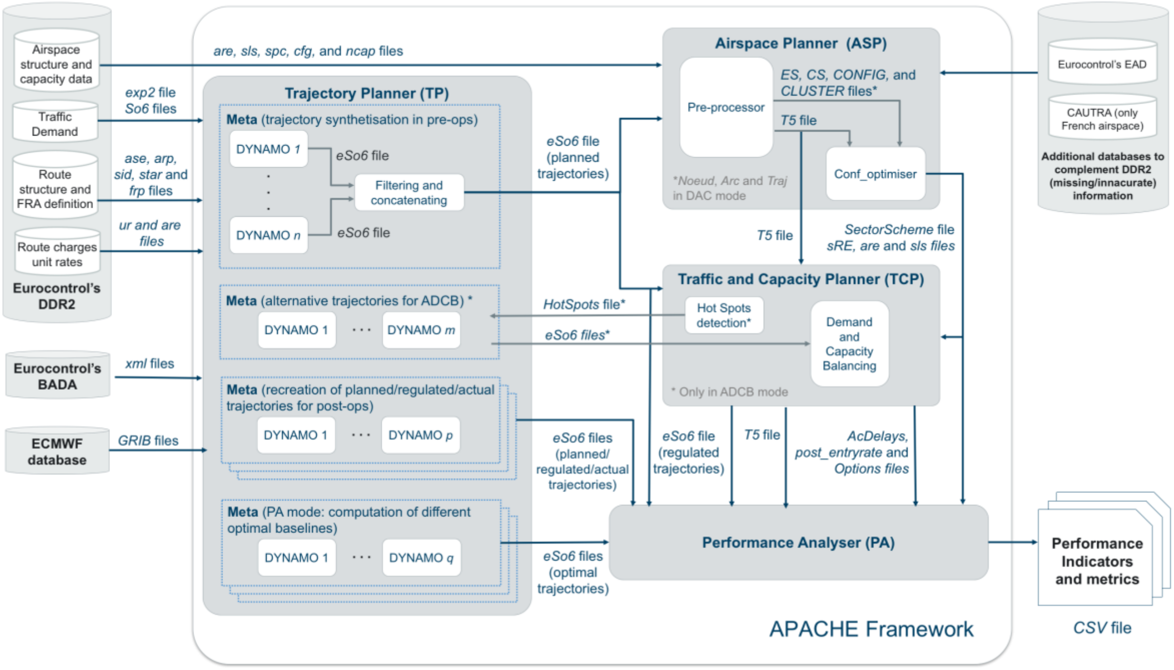


Figure 2-1. APACHE Framework workflow

The workflow starts by executing **Meta**, a component of the APACHE TP allowing a distributed computation of trajectories. See Appendix B for a detailed description of how high performance computing (HPC) techniques have been implemented for the TP in order to allow the massive computation of trajectories (around 1 million in the APACHE project).

2.1.1.1 Meta

Meta receives the traffic demand file (**exp2**) obtained from the Eurocontrol's Demand Data Repository (DDR2) for the selected period of time (typically one day) and geographic area (e.g. FABEC). The **exp2** includes, for each flight, basic information about the departure time, the origin and destination airports, a unique flight identifier, the callsign, the aircraft ICAO code (e.g. A320) and the requested flight level by the airspace user as submitted in the flight plan. It should be noted that the exp2 stores this high-level information, but not trajectories.

Based on the exp2 inputs file, meta generates two lists of flights: the **ignored flights** and the **flights to simulate**. Discarded flights will just be ignored by the APACHE framework and no trajectories will be simulated/optimised, neither taken into account by any performance indicator. These discarded flights are composed by:

- Flights with a **requested flight level lower than FL195** (one limitation of APACHE is that only the upper airspace is considered); and
- piston engine aircraft or helicopters (models not supported by current APACHE TP implementation).

All other flights will be considered by the APACHE TP, which will attempt to optimise or recreate their trajectories.

Besides the **exp2** input file, a detailed trajectory file is also needed in these particular situations:

- For Post-ops Case Studies, where fuel consumption, trajectory cost (among others) have to be estimated for certain Performance Indicators.
- To compute vertical flight inefficiencies only in the vertical trajectory profile, which require to fixing the planned/actual horizontal trajectory[2].
- Due to unreliable DDR2 database consistency outside the ECAC (regarding waypoints and segments), all portions of trajectory **outside the ECAC borders** will not be subject to optimisation for APACHE and the planned/actual route outside the ECAC (if any) will be fixed to that route initially planned/executed by the airspace user, being only the segment (or segments if the flight enters the ECAC many times) of **trajectory inside ECAC is subject to optimisation by the TP**. This has no impact in the performance assessment, since as explained in Section 2.1.4, the Performance Analyser will not account deliberately from flight inefficiencies outside the ECAC.

For this purpose, the Eurocontrol's DDR2 **So6** file will be used in APACHE. It is worth noting that any surveillance file containing trajectory data could be used here and perhaps with higher accuracy than **So6** files.

In the **So6** file, each trajectory is decomposed in several flight segments recorded by the different ATM units of the EUROCONTROL countries. Each line of this file corresponds to one segment and consists of 20 fields describing, among others, start and end latitude, longitude, flight level, date and time. In DDR2, three types of So6 files are found (Eurocontrol, 2016; Wandelt et al., 2014):

- a trajectory recreation based on the **last filed** flight plan by the airspace user (M1 file);
- a trajectory recreation based on the **regulated** flight plan (M2 file)[3]; and
- a trajectory recreation obtained from the position correlation from different surveillance systems (M3 file).

Based on the information included in these two files, Meta generates the input files for each individual flight to either predict or optimise the trajectory with DYNAMO. **Meta** splits the complete set of flights to simulate in as many chunks of flights as nodes available in the cluster (see Appendix B for details). Then, each node predicts or optimises the trajectories of its associated flights by executing DYNAMO in a parallelised manner. As a result of predicting (resp. optimising) a flight, DYNAMO generates an **extended So6 (eSo6)**, a file with detailed information about the predicted (resp. optimal) trajectory and a **log** for debugging purposes, which are dumped in the output folder of the corresponding flight.

This **eSo6** file is an extension of the generic Eurocontrol's original so6, specifically designed by the APACHE consortium and adopted as standard format to exchange traffic information between the different APACHE components. The format and information of the first 20 columns of the eSo6 are identical to those of the so6. In the eSo6, 7 additional columns are appended, which describe flight information required by the different APACHE components. This information includes: ground speed, track, rate of climb/descent, fuel consumption per segment, cost index, total route charges of the flight and trajectory identifier. The detailed specification is given in Appendix C of this Deliverable. It should

---

[2] See in (APACHE, 2018) the Environmental (ENV) indicators ENV-1.2, which fix the route distance of the RBT; ENV-2.1, which fix the trip fuel of the actual route; and ENV-2.4 and ENV-2-8 fixing the trip fuel of the RBT route.

[3] M1 and M2 trajectories are identical and might only differ in flight delay.

be noted that an *eSo6* can be easily converted to a generic *So6*, which is recognised by standard analysis and modelling tools such as NEST, by simply removing the last 7 columns.

Finally, once the chunks of flights of all the nodes have been executed, a **raw eSo6** is generated by **concatenating** the complete eSo6 (i.e. from origin to destination) of all the flights that have been simulated properly. Yet, the resulting eSo6 may include some outlier and/or incorrect trajectories coming from erroneous TP runs, which are mainly due to incorrect/corrupted inputs for the TP (e.g. a flight too short with a very large cost index that is not able to reach a flight level higher than FL195 as it was initially stated in the flight plan).

This **raw eSo6** is then **filtered by Meta** using data mining techniques, including principal component analysis (Jolliffe, 2014) and Mahalanobis distance criteria (Huberty, 2014), aiming to remove these outliers that may corrupt the results of the subsequent performance assessment.

All the other APACHE components downstream in the workflow will use this filtered eSo6. Detailed information about the number of flights to simulate, ignored flights, successfully executed flights and final number of flights after the outlier detection analysis can be found in the README file generated by the TP.

2.1.1.2 DYNAMO

In DYNAMO, the thrust and fuel flow for the different throttle settings and the drag coefficient for the different aircraft configurations rely on propulsive and aerodynamics models obtained from the Base of Aircraft Data v4.x (**BADA**) (Nuic et al. 2010), a database generated and maintained by EUROCONTROL, in cooperation with aircraft manufacturers and operating airlines, which contains aircraft performance models suitable for modelling and simulation of trajectories in support to various ATM research applications. BADA v4.x overcomes the known issues of BADA v3.x (Senzig et al., 2009) by providing enhanced models for the aircraft performance functions in the various flight regimes, and by considering compressibility effects in the aircraft drag coefficient model. In BADA v4.x, the performance functions are parameterised as a function of a set of aircraft-dependent coefficients. The coefficients for each particular aircraft model are specified in a separate XML formatted file. This file is determined by Meta for each flight based on the aircraft ICAO code obtained from the exp2, and is used to initialise the aircraft performance model in DYNAMO.

It should be noted that not all the aircraft ICAO codes appearing in the **exp2** have a known BADA model equivalent. In order to face this issue and compute the trajectories of as many flights as possible, a dictionary of synonyms (i.e. a dictionary that matches ICAO codes not present in BADA to the most similar known BADA model) has been generated using clustering techniques (Hartigan and Wong, 1979) based on the maximum take-off mass (MTOM), the wing span and the aircraft length. Accordingly, each time an ICAO code not known is detected, it is included to the dictionary of synonyms for future simulations.

In addition, DYNAMO can be fed with realistic weather data. These data must be provided in GRIdded Binary (**GRIB**) format, a concise data format used in meteorology to store historical and forecast weather data, composed by a collection of weather records defined at a regular grid of latitudes and longitudes for different pressure levels and times. A wide variety of GRIB files with diverse weather records, resolutions, time scales and accuracies are produced by the meteorological agencies. DYNAMO extracts the necessary weather data from these files using the ECMWF (European Centre for Medium-Range Weather Forecasts) GRIB-API, and approximates them by means of tensor product cubic splines, aiming at enabling their use during the numerical optimisation process. The GRIB file corresponding to each flight is determined by Meta, based on the departure time shown in the exp2.

DYNAMO also requires the route structure when restricting the lateral route to follow the current ATS route network (in addition of the definition of free route areas, if any). This information includes: airports, waypoints and airways interconnecting them, direct routes and free route areas. In addition, when optimising the lateral route considering the cost of the route charges, the unit rate cost of each country member of the ECAC area is required. All these data are obtained from the DDR2 of EUROCONTROL.

When DYNAMO is used to generate the alternative trajectories for the **TCP module in ADCB mode**, the list of airspace blocks to be avoided for each flight (if any) has to be specified, including the coordinates describing the boundaries and the lower and upper flight levels.

Finally, DYNAMO requires a configuration file, which specifies the lateral and vertical ConOps, the departure time, the payload and the Cost Index. This file is filled by Meta for each particular flight.

As a result of the prediction/optimisation process, DYNAMO generates an **eSo6** for the trajectory being predicted/optimised, a file with detailed information of the trajectory for debugging and plotting purposes, and a log which includes the steps carried out during the execution and eventual warning and errors.

2.1.2 Airspace Planner (ASP)

Next component in the APACHE workflow downstream, the Airspace Planner (ASP) takes input traffic file eSo6 (coming from the APACHE TP module) and airspace structure and capacity input files. The main output is the optimal sector opening scheme. Appendix A.2 details the format of the ASP input/output files.

The APACHE ASP is composed by two main software components: the data pre-processing module (**preprocessor**) and airspace configuration module (**conf_optimizer**), both being able to work in two modes representing current and future ConOps.

2.1.2.1 Preprocessor

For each simulated Case Study, the **preprocessor** is executed first and based on the input data; it computes constraints and objectives for the optimization problem. The **preprocessor** receives the traffic file (**eSo6**) obtained from APACHE TP module containing the lists of filtered flights for the given scenario, and for each flight list of trajectory points indicating time, position and speed vector.

Airspace structure and capacity data, as the second main input for the **preprocessor**, are gathered mainly from the Eurocontrol's Demand Data Repository (DDR2). Due to missing and wrong data in the DDR2 repository **additional sources** like Aeronautical Information Publication (AIP), available through Eurocontrol's European AIS Database (EAD), as well as French national aeronautical data repository (CAUTRA) and different ACC internal documentations were used to amend and complement DDR2 data.

Since all APACHE scenario included nominal conditions without simulating significant disruptive events (storms, military activities, strikes, etc.) it was necessary to extract nominal sector capacities. To achieve this**, it was not sufficient to use capacity data extracted from ncap file** for the particular date of the simulated scenario, since ncap file would contain only capacities of the sector that were active on that day and plus it will contain actual capacities that could be possibly affected by disruptive event that took place during that day. Hence, ENTRY_COUNTS file containing nominal sector capacities is computed using multiple AIRAC ncap data during 2015-2016 (from AIRAC 1501 to AIRAC 1613). Even then some sector capacities were still missing. Those capacity values were collected either from alternative sources (for the French airspace) or set manually to the closest value of the 'similar'[4] sectors.

Wrapping up, the input data list contains following files[5]:

- **are** Region file describing 2D piece of the airspace that represent polygon bases of the sector building blocks. The polygon bases are defined with list of latitude/longitude coordinates and linear segments connecting them.
- **sls** Sector list file describes airspace elementary sectors defined with set of 2D polygons (are file) with bottom and top level of the volume of the airspace they include.
- **spc** Airspace file defines collapsed sectors as a grouping of elementary sectors (sls file).
- **cfg** Configuration file defines all airspace configurations as set of sectors (sls and spc files) they include.
- Finally, **ncap** Airspace capacity file provides the actual capacity value of active sectors in the given AIRAC.

Based on these inputs, the **preprocessor** computes the airspace traffic intersection file in the NEST **t5** format. NEST file format is used to facilitate APACHE modules verification by enabling use of NEST tools. T5 Airspace Traffic intersection file is a NEST standard file used to represent traffic as a list of sectors flights traverse with entry/exit time, flight level and distance from last way point on the given route segment where entry/exit happened. For each traversed sector, total route distance and time spent in the sector are given as well. The file is produced using traffic input file **eSo6** and airspace structure files **are** and **sls** containing only elementary sector in the area of interest (France, FABEC, ECAC).

The **t5** output is further used by **preprocessor** in the current ConOps to compute sector entry counts, evaluate sector load and feasibility (whether load correspond to the capacity) and to finally evaluate airspace configuration feasibility and objective. The **t5** output is also used by **optimizer**, as well by other modules in the APACHE workflow downstream APACHE TCP (traffic and capacity planner) and PA (performance analyser). In the future ConOps mode (dynamic airspace configuration), the **preprocessor** computes traffic complexity for each trajectory point and matches them with airspace structure producing sector load in terms of traffic complexity and transfer traffic flows between neighbouring sectors.

The **preprocessor** has three additional roles to compute feasible configuration transitions, controller workload limit in the terms of traffic complexity and sector neighbouring:

---

[4] sectors consisting of same elementary sectors with difference of +/- 1

[5] the final list of required inputs depends of the Airspace Management ConOps of the simulated scenario

- Configuration transitions model operational constraints of sector grouping/degrouping in the Static sectorization problem. Feasible transition between two configurations signifies that those airspace configurations may be used in the consecutive period since active controllers could change sectors of their responsibility respecting comprehensive set of operational constraints. The configuration transitions guarantee that provided optimal sector opening scheme, as a result of ASP, could be used in the current operations.
- The controller workload limit in terms of traffic complexity represents sector capacity parameter in the Dynamic sectorization problem. In the APACHE project, it was decided to evaluate this parameter as the maximum complexity value of the active sector in the optimal sector opening schemes computed for the Static sectorization simulation runs.
- Finally, neighbouring between elementary sectors, which are used as Sector Building Blocks in Dynamic Airspace Configuration (DAC) mode (ASP in “future ConOps” mode) is used to ensure airspace continuum when sectors are grouped to form controllable sectors in DAC.

Taking in combination capacity and traffic input data, the **preprocessor** is computing matrices of the airspace configuration objective value and feasibility for each time period. Those matrices are fed into **conf_optimizer** as a principal input for the Static sectorization problem. TRANSITION_CONF, computed by **preprocessor**, contains for every predefined airspace configuration (cfg file) a list of configurations that represent feasible transitions.

Additionally, airspace structure input files are transformed by **preprocessor** to the ASP format known by the **conf_optimizer**:

- **ES** containing list of elementary sectors given by their unique id as in the **sls** file.
- **CS** containing list of collapsed sectors given by their unique id from **spc** file and for each list of elementary sectors that it contains.
- **CONFIG** gives a list of pre-defined airspace configurations with their unique id from cfg file, cluster/ACC it belongs, and for each list of sectors it contains.
- **CLUSTER** file lists all clusters/ACCs with id and descriptive name.

In the future ConOps mode (dynamic airspace configuration), the **preprocessor** computes, as a final output, following three files needed for the **conf_optimizer**.

- **Noeud** file defines all sector building blocks that are grouped by **conf_optimizer** to form controllable sectors. They are given by the id (same as id form sls file), x, y, and z coordinates of the barycentre and calculated complexity for each period.
- **Arc** file defines sector building block links, describing blocks neighbouring, that enables the **conf_optimizer** to choose feasible block groupings. Link’s transfer traffic flow for each period are also included in the arc file.
- **Traj** file contains for each flight list of sector building block traversed in each period, which is used to compute controllable sector convexity

2.1.2.2 Conf_optimizer input-output files

After the **preprocessor** has finished and inputs are computed, the **conf_optimizer** is launched generating an optimal sector opening scheme. The optimization problem of finding optimal sector opening schemes is modelled differently for the Static and Dynamic case, and therefore two different optimization technques are used to solve it. Static sectorization problem is modelled as Shortest path problem and solved using Dynamic programming (Cormen et al., 2009). On the other hand, Dynamic sectorization is modelled as Graph colouring and Heuristic method (Genetic algorithm) is used to solve it (Sergeeva et al., 2015; 2017).

The **conf_optimizer** generates different outputs used for module verification and validation, results analysis, etc. The most important is **SectorScheme** file that represents main input for the APACHE TCP (along with the traffic demand), as the next component in the APACHE workflow downstream.

Conf_optimizer in the **static sectorisation** mode (current ConOps) uses the following airspace structure input files in the ASP format computed by **preprocessor**: ES, CS, CONFIG, CLUSTER. All operational data regarding traffic demand, capacities, and other operational constraints are provided to the **conf_optimizer** in terms of configuration objective and feasibility matrices also computed by **preprocessor**.

Conf_optimizer in the **Dynamic Airspace Configuration (DAC)** mode uses three mentioned input files in the ASP format computed by **preprocessor**: noeud, arc, traj. The capacity value in terms of traffic complexity, also computed by the **preprocessor**, is an additional parameter needed by the **conf_optimizer**.

The **conf_optimizer** generates different outputs used for the verification and validation of the module, results analysis, etc. The standard output of the optimal opening scheme is given by **cos** file in the NEST format. It contains, for each time period, the list of active airspace configurations grouped by clusters/ACCs. The cos file facilitates ASP solution analysis by importing it in the NEST tool, however it does not contain sufficient information needed by PA to compute performance indicators. The **sRE** file enriches this information providing, for each time period, the number of open positions with list of active sectors. Detailed sector opening scheme output file further enriches available information adding sector load for each active sector and grouping them by cluster/ACC. This file contains all information needed for result analysis and comparison of different solutions, and it is used during model validation.

The most important output file, **SectorScheme** (used by TCP), combines the resulting opening scheme with airspace structure and capacity information enabling TCP module to use it as single source (in addition to t5 traffic demand file). For each elementary sector, it lists active sectors[6], to whom that elementary sector belongs at the given time period, with capacity information of the active sector. Detailed format description of **SectorScheme** file is given in the Appendix D.

2.1.3 Traffic and Capacity Planner (TCP)

As reported in D3.2 (APACHE Consortium, 2018), TCP has two modes of execution:

- "current ConOps" replicating the Computer Assisted Slot Allocation (CASA) algorithm, assigning delays when a demand and capacity imbalance occurs; and
- "future ConOps" implementing an advanced demand and capacity (ADCB) algorithm, allowing for optimal delay and trajectory amendments at pre-tactical level).

---

[6] elementary or collapsed that were part of the optimal sector opening scheme.

Both modes of operations use the same inputs (Appendix A.3 details the format of the TCP input/output files):

- From the Airspace Planner (ASP) the TCP gets its main output: the **airspace configuration** and opening scheme (SectorScheme file as detailed in Appendix D). This file has the description of which sectors are active per period of time during the day. The periods of activation of the sectors are given in 20 minutes or multiples of 20 minutes. The format of the file containing the airspace configuration was designed *ad-hoc* for the purposes of this Project.
- Another input file Airspace/Traffic intersection file, also known as **T5 file**, is also shared by ASP. This file provides the intersection time (entry time and exit time) of the all trajectories and all the basic sectors. The NEST input is the set of trajectories given by the TP in So6 format. The T5 file format is the same as provided by NEST with no modifications.
- From the Trajectory Planner (TP) the TCP obtains the eSo6 file containing all the trajectories. This file is in fact not needed to calculate the delays in CASA mode, nor to detect the hostspots in ADCB mode (it is done from the T5 file). In ADCB mode, however, this file is needed to compute the along-path distances of the intersections between the hotspots (congested sectors) and the concerned trajectories. Further more, in both modes this *eSo6* file is always read and saved as a new eSo6 file (main TCP output file) update the trajectory information for the downstream components of the APACHE System, indicating for instance, which trajectories have been regulated.

Additionally to the input files the TCP has also some parametrization. In CASA mode, there are two main parameters, one to set the percentage of the allowed capacity overload (set to 20% to all WP4 Test Cases), and the number of iterations the algorithm is run (set to 1 to all WP4 Test Cases). In the ADCB mode the only parameter is the percentage of the allowed capacity overload (also set to 20% to proper comparisons).

The main output file of the TCP is a **new eSo6 file (regulated flights)**. This file contains the same amount of flights as the input eSo6 file provided by the TP (planned flights). The update done by the TCP is to add the delay (if any) to the concerned flights, by changing the time columns of each segment of the trajectory of the input eSo6 file. Moreover, in ADCB mode, the route or the vertical profile is also changed for those flights that a tactical re-routing or level capping (respectively) has been chosen by the ADCB algorithm.

This *eSo6* output file is also updated to visualise which trajectories have been modified by the TCP (i.e. regulated). The last column of the file is changed from SBT_x to RBT to indicate this trajectory is now the reference business trajectory. Moreover, if the trajectory has been updated by the TCP then an asteric character is added (RBT*) to indicate that a regulation is indeed applied. This regualtion could be simply delay (after a demand and capacity balance using CASA or might involve re-routings or level cappings (see Appendix C of this document for details on the *eSo6* format).

Another important output file for TCP in mode ADCB is the *HotSpots* file. This is a compressed set of files (one per flight crossing a hotspot), which contains information on the involved flights and the geometric characteristics of the sector to be avoided. This is the main interface file for the backwards interaction between the TCP and TP, in ADCB mode. In this second execution the TP will compute alternative trajectories (lateral or vertical re-routings) for each concerned flight avoiding the listed hotspots provided in these files.

2.1.4 Performance Analyser (PA)

The PA module receives the output files obtained from **APACHE-TAP** components and calculates the different performance indicators (PIs), according to the specifications provided in deliverable D3.2 (APACHE, 2018). It is implemented in such a way to automatically choose a set of PIs to be calculated, depending on the type of assessment ("pre-ops" or "post-ops"). Moreover, there is a possibility to manually choose the necessary PIs by the user.

The main input files, used for the calculation of the majority of PIs, are: planned trajectories coming from TP and regulated trajectories coming from TCP; both in the aforementioned **eSo6** format. For the **pre-ops scenarios** (synthesised scenarios) both files are used. For **post-ops** scenarios, only eSo6 coming from TP is used since the planned, regulated and actual trajectories, taken from DDR2 (from M1, M2 and M3 files, respectively), are reconstructed by the TP.

As depicted in Figure 1-3 the Performance Analyser implements some complex metrics that require inputs from optimisation tools provided by the APACHE-TAP, such as optimal trajectory baselines or optimal sectorisations. These are respectively provided by TP (in eSo6 format) or ASP (in SectorScheme format), when properly configured in "**PA mode**" (i.e. computing optimal baselines and not synthesising traffic/sectors to create a given scenario).

As seen in Figure 2-1 the PA also needs the *T5*, *ACDelays*, *post_entryrate* and *Options* files (from the TCP) and the *sRE*, *are* and *sls* files from the ASP to compute certain Performance Indicators.

The PA embeds the **Risk Assessment (RA) module**, which is a specific module, developed entirely by UB-FTTE, that computes all SAF (safety) Performance Indicators from the input eSo6 files.

Finally, it should be noted that the APACHE project aims at assessing ATM performance at ECAC level, thus inefficiencies of the ATM system outside the ECAC should not be captured. If the trajectory of flights with origin and/or destination airports outside the ECAC were optimised from origin to destination, ATM inefficiencies before entering and/or after leaving the ECAC would be quantified and could obscure the results of the performance assessment. Thus, **all trajectory segments outside the ECAC will not have influence on the performance metrics** (even if those segments are taken into account when generating or reconstructing the trajectories).

The PA results per each PI are stored in two output files *(_output.txt* and *_debug.txt).* These files are then post-processed in order to obtain a consolidated CSV (comma separated value) table, merging into a single table all PIs and metrics for all KPA and Case Studies, which will be used as main input in WP5 for performance assessment, benchmarking and visualisation purposes. Appendix A.4 details the format of these output files and final CSV table.

2.2 APACHE shared file system

As commented before, although the different software modules that compose the APACHE System are located in different premises, a shared file system has been created in order to centralise all relevant data used by the APACHE System, and to facilitate the interchange of information among the different components according to the workflow explained above.

For this purpose, a dedicated Linux server at UPC premises has been setup, accessible by all APACHE consortium Partners via secure shell (SSH) connection. Within UPC, the different machines dedicated to run the TP in a distributed mode have also access to this file system using the NFS protocol (Network File System).

The shared file system is structured in three main folders:

- **/mnt/apache/ddr2_inputs** [*read-only*], containing the original data as downloaded from Eurocontrol's Demand Data Repository 2 (DDR2): traffic demand and route network, ATC sector information (definition of sectors and nominal capacities) of the applicable AIRAC cycle(s). This data is used as main input to build the different pre-ops and post-ops case studies of the APACHE Project.
- **/mnt/apache/pru_inputs** [*read-only*]: containing original data provided by Eurocontrol's Performance Review Unit (PRU): Correlated Position Report (CPR) messages, which will be used as alternative trajectory data source in the post-ops Scenario of the Project.
- **/mnt/apache/scenarios** [*read-write*]: containing the data of all simulations.

The folders *ddr2_inputs* and *pru_inputs* have *read-only* permission in order to ensure that all simulations done in WP5 are using the same input data and to prevent accidental removal or editing of this data.

The folder *scenarios,* in turn, is organised as follows:

<Scenario>/<Area>/<Case-Study>/<Component>/<Mode>

In Table 2-1 a description of each field of this folder structure is given.

For the post-ops Case Studies, the same folder structure applies with the addition of a new folder (named *ATC*) that contains the actual flown trajectories. Recall that one of the APACHE System limitations is that tactical operations are not simulated in the pre-ops scenarios and therefore this folder does not appear for the pre-ops Case Studies.

Thus, for the **post-ops Case Studies** the folder structure is as follows:

- <Scenario>/<Area>/<Case-Study>/**TP**/: folder with the trajectory recreation of **the last filed flight plans** (from DDR2 M1 file) using the trajectory reconsctruction and estimation capabilities of the APACHE TP. This folder also contains the different baseline optimal trajectories (also computed by the APACHE TP) that are required to compute certain performance indicators. These different <modes> are stored here like in the pre-ops Case Studies.

- <Scenario>/<Area>/<Case-Study>/**ASP**/: folder with the actual **opening scheme** (from DDR2 files). This folder also contains the different baseline optimal sector schemes (computed by the APACHE ASP) that are required to compute certain performance indicators. These different <modes> are stored here like in the pre-ops Case Studies.

- <Scenario>/<Area>/<Case-Study>/**TCP**/: folder with the trajectory recreation of the **regulated flight plans** (from DDR2 M2 file) using the trajectory reconsctruction and estimation capabilities of the APACHE TP.

- <Scenario>/<Area>/<Case-Study>/**ATC**/: folder with the trajectory recreation of the **actual trajectories** (from DDR2 M3 file) using the trajectory reconsctruction and estimation capabilities of the APACHE TP.

- <Scenario>/<Area>/<Case-Study>/**RA**/: folder with the results of the SAF specific performance indicators (like with the pre-ops Case Studies).
- <Scenario>/<Area>/<Case-Study>/**PA**/: folder with the results of all other performance indicators (like with the pre-ops Case Studies). APACHE System input-output and interface files

Folder	Description	Possible values
<Scenario>	Scenario folder, which may contain several case studies.	*S0, S1, S2, S3, S4, S5, S6, S7*
<Area>	Area folder in order to differentiate case studies done at different geographic scopes.	*ECAC, FABEC, FRANCE*
<Case-Study>	Case study folder, defined by the day under assessment. Since one of the objectives of APACHE is to assess KPA interdependencies and Pareto optimality, specific trade-off sensitivity studies will be stored in different folders.	*<Simulation-day>_<Start-time>_<Time-period>_<nnnnn>_[<Pareto-Front-option>]* See Table 2-2 for its specification.
<Component>	For each case study, a specific folder is used to store input/output information for each APACHE Framework component.	TP, ASP, TCP, ATC*, RA or PA * Only in S0 (post-ops), see below.
<Mode>	As commented before and seen in Figure 1-3., the APACHE Framework has a double functionality (synthesise scenarios or support the implementation of novel ATM PIs). Moreover, the TCP (in ADCB mode only) will request extra trajectories that simulate ATFM re-routings or level capping.	*Original:* This is the default mode and means the nominal simulation to synthesise the traffic of a given case study. *PA-PRE-xxx:* specific optimisation for the PA in pre-ops mode *TCP-LAT:* specific optimisation for the TCP implementing ATFM re-routings.

Table 2-1. Scenario folder naming convention

Folder	Description	Possible values
<Simulation-day>	Day taken for the simulation	in yyyy-mm-dd format
<Start-time>	Within the simulation day, at what time the simulation starts	From 00 to 24
<Time-period>	Within the simulation day, how much time (in hours) is simulated	From 00 to 24.
<nnnnn>	Number of flights of the simulation (for informative purposes)	Integer
<Pareto-Front-option>	Optional field in case this particular simulation corresponds to one of the Pareto assessments.	Therefore, this field could be PF1a, PF1b, etc.

Table 2-2 Case-study folder naming convention

In Figure 2-1 the system interface files are identified, along with additional output files of each component used for internal debugging or detailed results logging. A README file is also provided by each component summarising the status of the simulation, number of flights processed, execution time, simulation warnings, etc. Appendix A of this document enumerates all these files and details their content and purpose.

3 Verification and validation of the APACHE System

A proper validation of the developed model is a prerequisite in order to establish confidence in it. In (Sargent, 2009) and (Balci, 1998) model validation is defined as ‘‘*substantiation that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model*’’. Following (Balci, 1998), the main principles of validation are (Netjasov et al, 2013):

- Validation must be conducted throughout the entire life cycle of a simulation study.
- The outcome of model validation should not be considered as a binary variable where the model is absolutely correct or incorrect.
- A simulation model is built with respect to study objectives and its credibility is judged with respect to those objectives.

Since a model is an abstraction of a system, perfect representation is never expected. The outcome of the model validation should be considered as a degree of credibility on a scale from 0 (absolutely incorrect) to 100 (absolutely correct) (Balci, 1998).

The following definitions are commonly used (see Figure 3-1) as given (MITRE, 2014):

- *Verification*: “The process of determining that a model implementation and its associated data accurately represent the developer’s conceptual description and specifications.”
- *Validation*: “The process of determining the degree to which a [simulation] model and its associated data are an accurate representation of the real world from the perspective of the intended uses of the model.”
- *Accreditation*: “The official certification that a model, simulation, or federation of models and simulations and its associated data are acceptable for use for a specific purpose.”

This chapter firstly presents the results of the APACHE System integration and **verification** tests. These tests were used to evaluate the compliance with the APACHE System requirements that were drawn in Deliverable D3.2 (APACHE Consortium, 2018). Appendix E details this evaluation of these requirements.

Regarding the **validation**, it should be noted that this Deliverable presents the validation at **component level** (i.e. the TP, ASP, TCP and RA independently). It is out of the scope of this document to validate the whole integrated APACHE System or Framework. This will be subject of the APACHE validation activities foreseen in WP5 and will be reported in APACHE Deliverable D5.1.

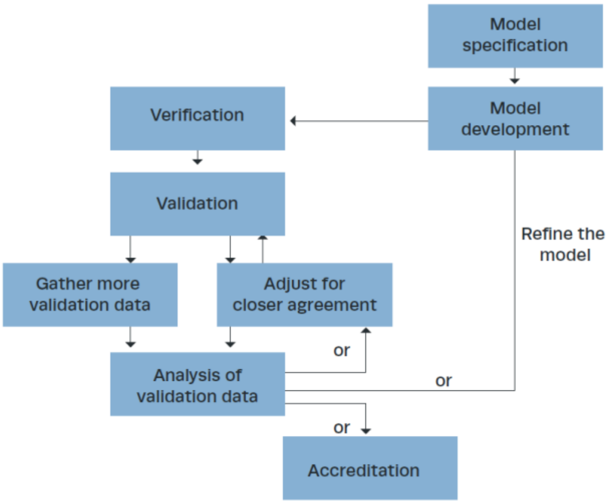


Figure 3-1. Simulation model development with the verification, validation and accreditation processes (MITRE, 2014)

3.1 APACHE System integration and verification results

As detailed in Table 3-1, **four integration and verification tests**, involving the full workflow of the APACHE System (see Section 2.1) were planned. In the four test cases the APACHE-TAP was used to synthesise trajectories and airspace configurations for "pre-ops" assessment purposes, and to support the implementation of some selected Performance Indicators. The idea of these Test Cases was to test all the APACHE-TAP components in their two possible modes (i.e. "current ConOps" and "future ConOps").

It should be noted, however, that Test 4 was not finally performed since the DAC mode of the ASP was not finally integrated into the APACHE workflow (see section 3.1.2.2).

Test Case	TP *original* mode	ASP *original* mode	TCP *original* mode
Test 1	**"Current ConOps":** en-route networks and FL allocation/orientation schemes	**"Current ConOps":** Static sectorisation	**"Current ConOps**": Computer assisted slot allocation
Test 2	**"Future ConOps":** Full free route and current FL allocation/orientation schemes	**"Current ConOps":** Static sectorisation	**"Future ConOps":** Advanced demand and capacity balancing
Test 3	**""Current ConOps":** en-route networks and FL allocation/orientation schemes	**"Current ConOps":** Static sectorisation	**"Future ConOps":** Advanced demand and capacity balancing
Test 4	**"Future ConOps":** Full free route and current FL allocation/orientation schemes	**"Future ConOps":** Dynamic Airspace Sectorisation (DAC)	**"Future ConOps":** Advanced demand and capacity balancing

Table 3-1. Description of the APACHE System integration and verification tests

The traffic demand and AIRAC cycle is taken from **February 20th 2017**, during **24h** and only considering those flights crossing the **French airspace**. Demand data has been obtained from Eurocontrol's DDR2, including the aircraft type, departure time and origin/destination airports. Airspace data, consisting of elementary/collapsed sector and airspace configurations definition, as well as, capacities of the sectors; were also taken from the AIRAC data from the DDR2 supplemented by French national data repository.

In order to run the TP, weather data for the same day and region of study was gathered from the Global Forecast System (GFS), a weather forecast model produced by the National Centers for Environmental Prediction (NCEP) and provided in GRIB formatted files. Aircraft performance data, for each aircraft type, was obtained from Eurocontol's BADA v4.2.

DDR2 files contained an initial demand of 7,375 flights. Nevertheless, since the APACHE Project focuses in the en-route phase, all flights with a requested flight level below FL195 were discarded for the simulations. Moreover, helicopter and piston engine aircraft were also discarded, leading to a total of 6,895 scheduled flights analysed in this test case.

3.1.1 Trajectory planner (TP) component

For the three integration and verification tests aircraft trajectories have been synthesised using the APACHE TP component, taking into account the input demand from DDR2 (origin/destination airports, aircraft type and take-off time). For this purpose, each flight has been simulated with a random Cost Index (CI) and landing mass following normal distributions.

On the one hand, the normal distribution for the CI has been derived empirically for each aircraft model. According to (Roberson et al., 2008) the typical cruise speed is that of Long Range Cruise (LRC). Nowadays, LRC speed is almost universally higher than the speed that would result from using the CI selected by most carriers. Based on this assertion, the CI leading to a cruise Mach number corresponding to that of LRC has been computed off-line for different flight conditions (aircraft mass, altitude, longitudinal wind and temperature deviation with respect to ISA). Then, the resulting experimental distribution of LRC CIs has been fitted with a Gaussian function, quantifying in this way the mean and standard deviation parameters. These parameters for each aircraft model are stored in a dictionary that translates an ICAO code to the corresponding Gaussian function description. During the process of DYNAMO inputs files generation, the Meta component of the APACHE TP selects a random CI for each flight, based on the Gaussian distribution corresponding to its aircraft model.

On the other hand, the normal distribution for the landing mass is centred to 90% of the Maximum Landing Mass (MLM), regardless of the aircraft model, with a standard deviation of 10%. When generating the random CI and landing mass, the unique flight identifier is used as a seed.

As shown in Table 3-1, two sets of trajectories were generated for the APACHE verification and integration tests: optimal trajectories constrained by current route network and current FL allocation/orientation schemes; and optimal trajectories assuming a full free route scenario (from origin to destination) and still constrained to current FL allocation/orientation schemes.

The input flights for the TP after removing helicopters, piston engine aircraft, unknown aircraft models and flights whose requested flight level below FL195 are 6,895. From this set of flights to simulate, the TP successfully generated 6,760 and 6,761 trajectories for the structured and free route scenarios, respectively. The concatenated trajectories of these flights compose the raw eSo6 (see Section 2.1.1). After removing potential outliers, the final set of flights is 6,733 and 6,761 for the structured and free route scenarios, respectively. The concatenated trajectories of the filtered flights compose the output eSo6, which is the TP System interface file that will serve as input by the ASP, TCP and PA, downstream in the workflow.

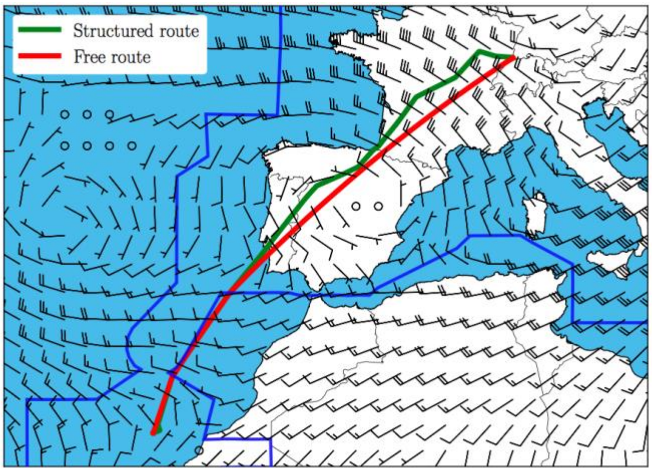
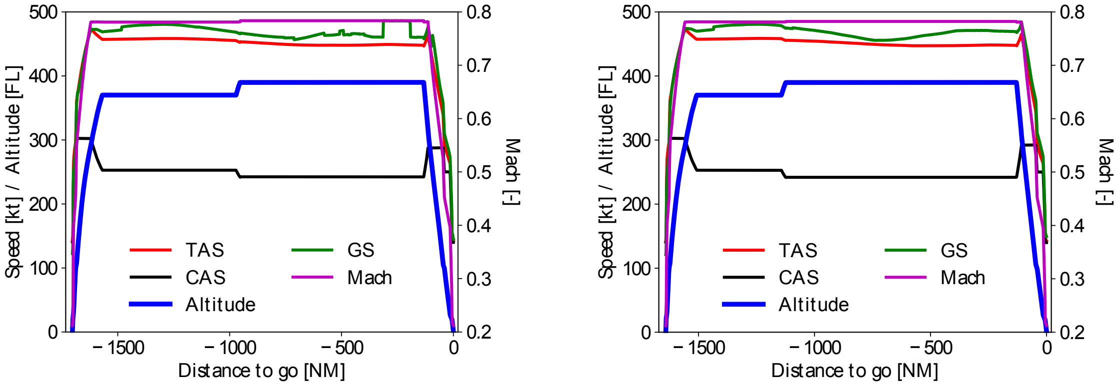


Figure 3-2. Lateral profile for the example trajectory



(a) Structured route (b) Free route

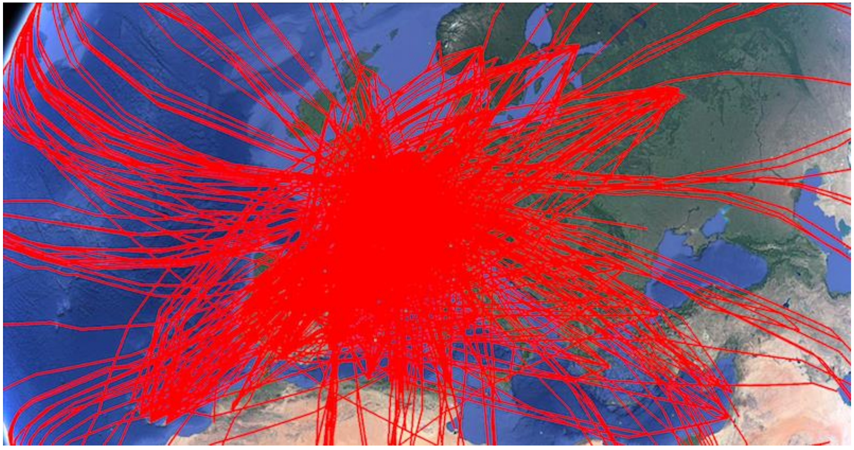
Figure 3-3. Vertical profile for the example trajectory

Figure 3-2 shows the lateral route for the two tests for an example flight of the input demand. The limits of the ECAC area are shown as a blue-solid line, and the wind barbs indicate the wind filed to a pressure altitude of 200 hPa (around FL380). As it can be observed in the Figure, the segment of route outside the ECAC is identical for both free route and structured route operations. This is because in APACHE trajectories are not optimised outside the ECAC borders and what is given in the input So6 file is fixed (see section 2.1.1). As expected, in a full free route operational context flights take more direct routes, since they are not restricted to fly along the published waypoints and airways of the ATS route network. In addition, this gives more freedom to follow favourable winds and maximise the ground speed.

Figure 3-3(a) and (b) show the vertical profile for the same trajectory under structured and free route operational contexts, respectively. The differences in the route lead to differences in the weather (wind, temperature and pressure) conditions found along the trajectory and, consequently, differences in the optimal vertical profile (it should be noted, for instance, that for the full free route trajectory the step climb is performed earlier than for the structured route trajectory).

Finally, Figure 3-4 show the complete set of trajectories simulated for the APACHE verification and integration tests. As expected, the spatial distribution for the full free route scenario is larger than for the structured route case, showing also more direct and efficient trajectories.

(a) Structured route



(b) Free route

Figure 3-4. Complete set of trajectories in the traffic scenario

3.1.2 Airspace planner (ASP) component

The airspace planner ASP was implemented according to the specification given in APACHE Deliverable D3.2 (APACHE Consortium, 2018). For the three integration and verification tests flight input data were provided by the TP module as indicated in the APACHE workflow (see Figure 2-1). The remaining input airspace structure and capacity data are mainly taken from DDR2 repository or computed by the ASP preprocessor component as explained in Section 2.1.2.

3.1.2.1 Verification of the ASP in static mode (current ConOps)

For the verification test the French airspace is taken as geographical scope. The French airspace (LF) is organized in the five area control centres – ACC (see Figure 3-5) each containing one or more airspace clusters:

- Bordeaux (LFBB) – 1 cluster.
- Brest (LFRR) – 3 clusters.
- Marseille (LFMM) – 2 clusters.
- Paris (LFFF) – 2 clusters (LFFFCTAA – approach excluded in respect with APACHE assumptions).
- Reims (LFEE) – 3 clusters.

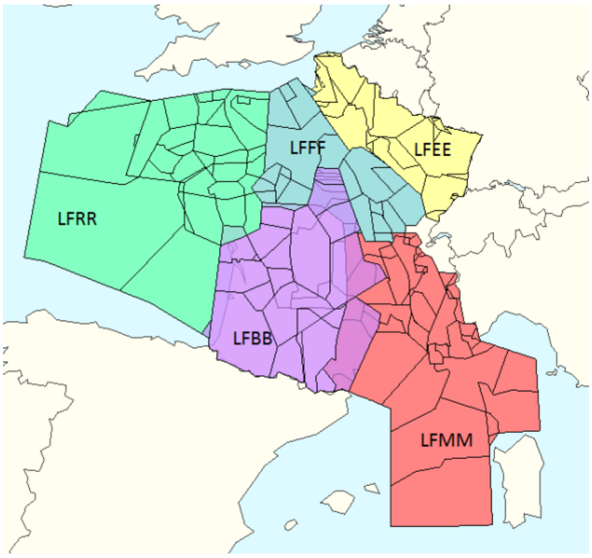


Figure 3-5. French airspace area control centres (visualized by Eurocontrol NEST)

Therefore, in total 11 clusters were optimized containing: 164 elementary sectors, 420 collapsed sectors and 1506 airspace configurations.

The ASP module was run for two sets of the traffic data: one using structured route (Test 1 and 3) and another using a free route scenario (Test 2), in the search for the optimal sector opening scheme that minimizes number of active sectors and balance sector load among active sectors.

Active sectors of the Test 1 at two time instances (9h and 11h), that are part of the optimal sector opening scheme, are visualized in the Figure 3-6, which depicts the change of active sectors (in number and/or redistribution of elementary sectors) due to change in the traffic demand. Due to increase of traffic demand at 11h compared to 9h, degrouping (splitting) of collapsed sectors into smaller sectors is mainly visible in the Figure 3-6. Similar results are found for Test 2.

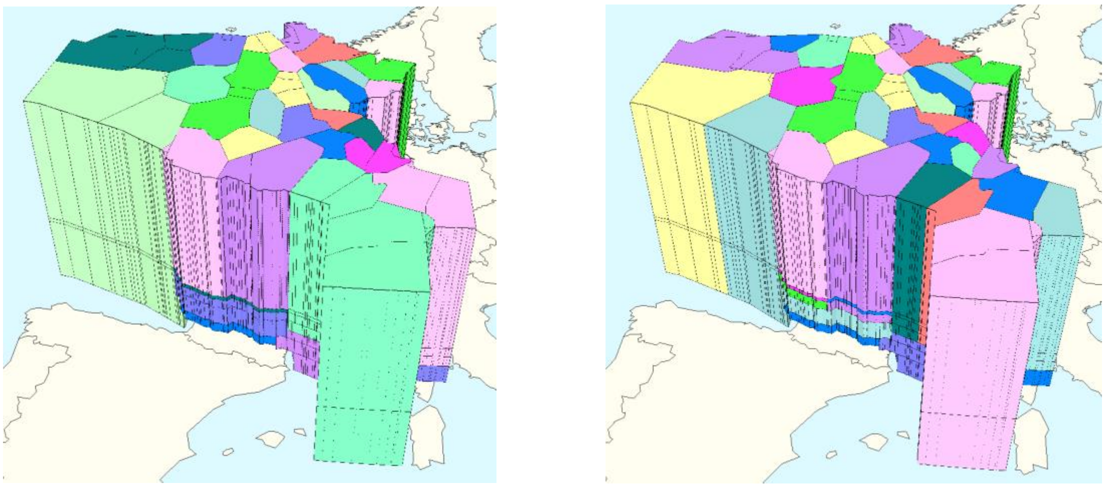
Figure 3-7 shows the distribution of the number of active sector and Figure 3-8 shows the five-number statistic distribution of the sector loads for both tests for the pick period of the day from 9h till 13h. Although it was not a purpose of the verification tests, a quick view on the results already reveals that optimal sector opening scheme for the traffic that uses structured route yield slightly lower number of active sectors than optimal scheme for the traffic that uses free route.

But what is more important, this graph shows that number of active sectors for both tests are constantly changing, adapting to ever changing traffic conditions[7] (yellow line in the figure). With the increase of traffic demand, which certainly causes the increase of sector loads, the Figure shows that the number of active sectors is also increased, confirming that algorithm is splitting collapsed sectors in more sectors in order to keep sector load below allowed limit (capacity). Conversely, when traffic demand is decreasing, sectors are grouped in fewer sectors maximizing use of the sector capacity and

---

[7] For aggregation purposes, the traffic change is represented by entry counts (number of flight entering the airspace) of the whole French airspace. Although it may give general idea about traffic movements, it does not give sufficient information about entry counts (i.e. load of the sectors), and therefore traffic line and active position bars may not be fully aligned.

therefore minimizing number of active sectors. In the early morning and late evening, when traffic demand is lowest (23h – 4h), there is only one active sector per ACC in the resulting optimal opening scheme (not sown in the Figure 3-7).



(a) Airspace configuration at 9h

(b) Airspace configuration at 11h

Figure 3-6. Active sectors of the optimal opening scheme at two time instant (visualized by Eurocontrol NEST)

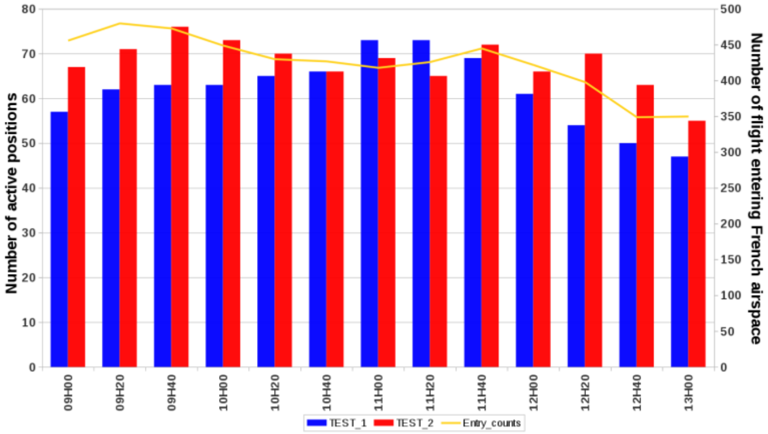


Figure 3-7. Distribution of the number of active sector

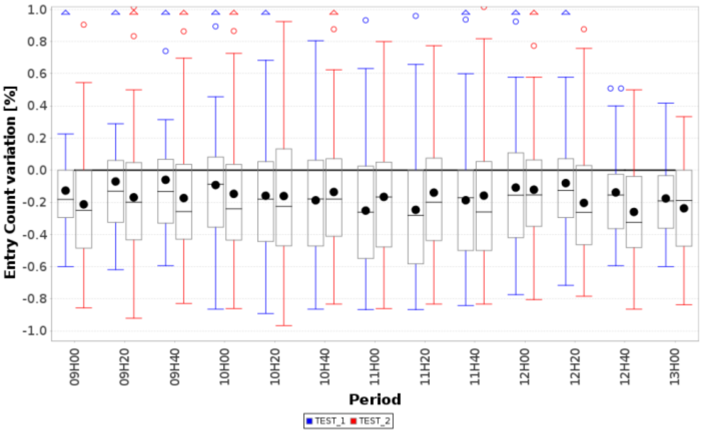


Figure 3-8. Five-number statistic distribution of the sector loads

This verifies the main function of the ASP module, which is to minimize the number of the active sectors, while fulfilling all operational constraints. See also Figure 3-8, where the mean sector load (black point in the figure) has almost a constant small negative value (underload) meaning that sectors are almost always fully loaded, no matter traffic decreases or increases. Such high sector load guarantees a minimum number of active positions in the resulting opening scheme, which is a purpose of the ASP model. Furthermore, the distribution of the sector loads shows that interquartile ranges (black boxes in the Figure) or midspread values of the sector loads almost completely fall in the negative part of the load axes (underload), confirming that sectors are generally not overloaded and that sector capacity constraints were respected by the algorithm.

Figure 3-8 also reveals that maximum sector loads for each period are greatly positive which significate high sector overload. Although small in numbers, such sector overloads represent violations of the sector capacity constraints and the reason for such behaviour should be further analysed. Simple test of the active sector loads showed this happens because of high traffic demand at lowest controllable sector[8] level (elementary sector level), when elementary sectors, that may not be further separated, are left overloaded. This is not a rare situation in operations, when in the next phases of the ATFCM process, if the capacity may not be adapted, those sectors will cause a regulation of the demand. Table 3-2 confirms that only elementary sectors are left overloaded by the algorithm, verifying that algorithm always respect the sector capacity constraint.

Time	Sector	Entry counts	Capacity	Time	Sector	Entry counts	Capacity	Time	Sector	Entry counts	Capacity
1487583600	LFEEKD	87	36	1487586000	LFEEKD	76	36	1487572800	LFEEKD	78	38
1487582400	LFEEKD	86	36	1487571600	LFEEHYR	80	38	1487581200	LFEEHYR	67	33
1487582400	LFEEKF	76	33	1487592000	LFEEHYR	80	38	1487581200	LFEEKF	73	36
1487584800	LFEEKD	81	36	1487583600	LFEEKF	69	33	1487620800	LFEEKD	73	36
1487592000	LFEEKD	79	36	1487582400	LFEEKF	79	38	1487619600	LFEEKD	77	38
1487590800	LFEEKD	78	36	1487583600	LFEEHYR	79	38	1487593200	LFEEHYR	76	38
1487584800	LFEEHYR	81	38	1487571600	LFEEHYR	74	36	…	…	…	…

Table 3-2. Example of unsolved sector overloads for Test1

Further investigation reveals that mainly high elementary sector in the top-upper airspace are overloaded that is caused by the optimal choice of the cruising altitude without any ATM restrictions (see Figure 3-9). This Figure clearly shows that altitude limitations are imposed in the DDR flight plans, in order to reduce the load of high sectors that are reserved for over-flights. These altitude limitations are typically[9] specified in the Route Avalability Document, which restricts for instance certain cruise flight levels depending on the origin or destination airports, airway in use, etc. These flight plan

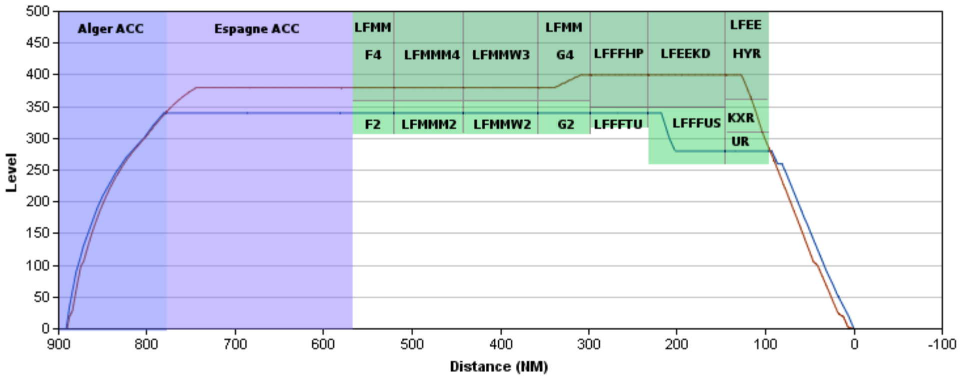
---

[8] For the sake of simplicity in the text we were referring to elementary sector as the lowest controllable airspace. However, in real operations, some elementary sectors are not controllable and smallest controllable sector in those cases are collapsed sectors, usually containing two elementary sectors. Those sectors are treated by algorithm in the same way, and in the case of high traffic demand left overloaded. An example is sector LFEEKD that is formed of two non-controllable elementary sectors LFEEHD and LFEELD.

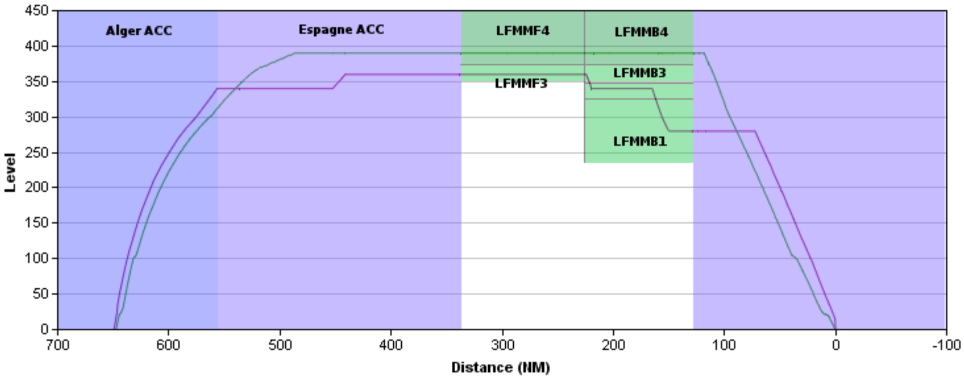
[9] These could also be the consequence of a level capping (altitude regulation at pre-tactical level, during the ATFM process).

restrictions are not modelled by the TP (see setion 4.2.1) and therefore, all trajectories comming out from this module follow optimal vertical profiles. This is seen in the Figure, where for the same example flights the TP chose a higher cruising altitudes, causing high demand for the high altitude sectors.

Figure 3-10 presents a detailed analysis of elementary sectors loads in the Bordeaux ACC, which compares sector loads from simulated traffic (that uses optimal cruising altitude) and DDR flight plans (including all ATFCM restrictions). The Figure shows north and south elementary polygons each divided in four vertical layers representing four elementary sectors[10], high altitude sector being at the top. Then for each elementary sector, a graph representing the distribution of the sector load from 10h to 12h (vertical axis) is given. This Figure clearly confirms the statement of high sector overload in the top airspaces due to optimal cruising altitudes. This without any doubts verifies that resulting sector overloads in the optimal opening scheme are not caused by the ASP algorithm but due to the TP limitation, which does not take into account possible altitude restrictions for certain routes and always selects the best cruising altitude.



(a) Example 1. In blue the DDR2 trajectory and in red the TP optimal trajectory



(b) Example 2. In red, the DDR2 trajectory and in green, the TP optimal trajectory

Figure 3-9. Effect of the optimal vertical profile on sector load and comparison with trajectories extracted from DDR2. Purple airspace is out of the geographical scope of the verification test, while green airspace represents sectors of interest for the verification presented here. Darker sectors are higher altitude sectors.

---

[10] For example polygon LFBBR in third layer with altitude ranging from FL345 to FL365 forms elementary sector LFBBR4.

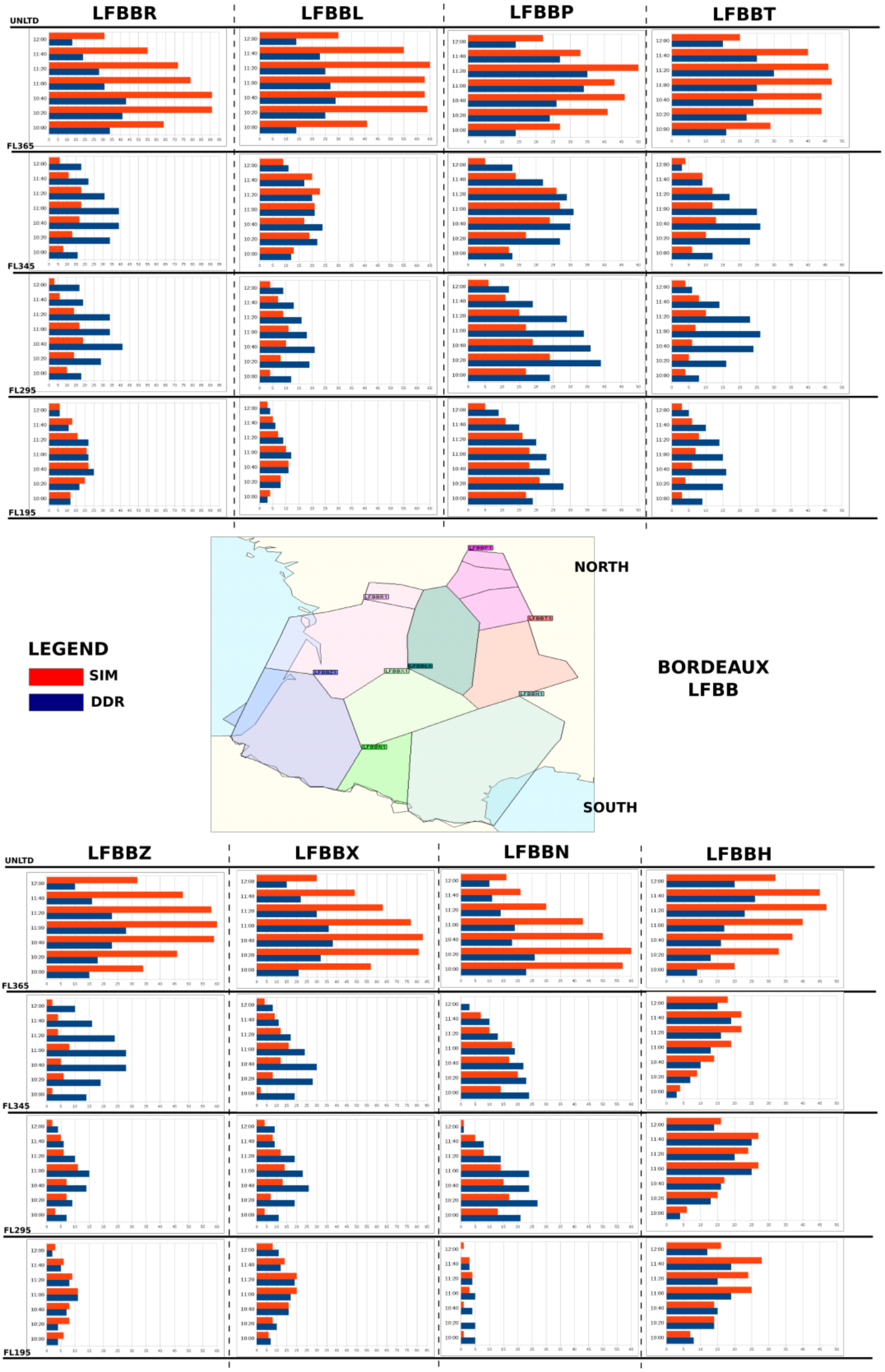


Figure 3-10. Detailed sector load analysis – Bordeaux ACC

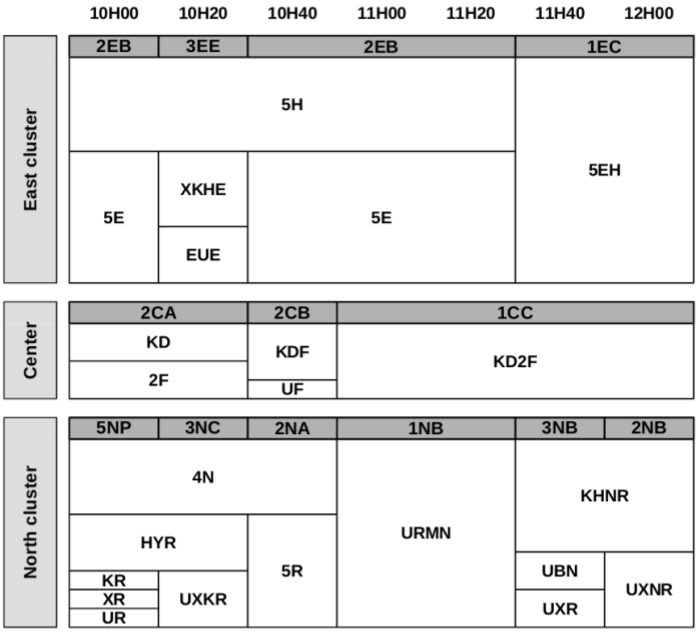


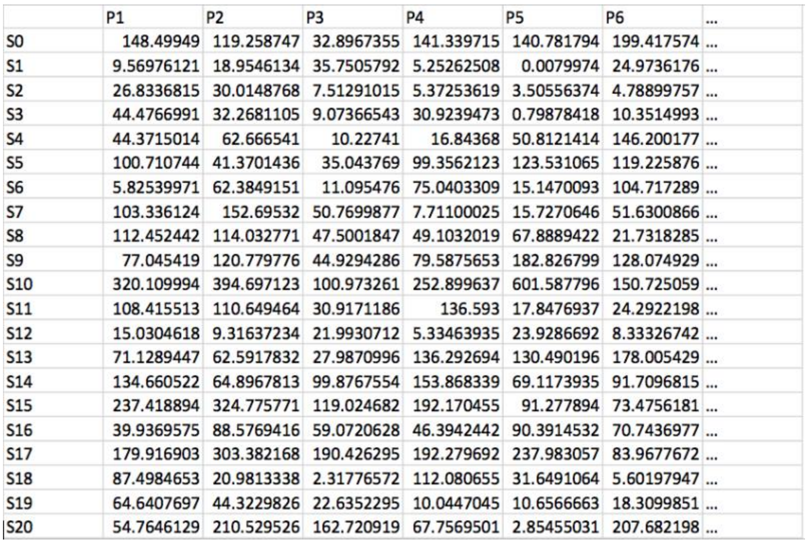
Figure 3-11. Sector opening scheme – Reims ACC

In the Figure 3-11, part of the optimal sector opening scheme of the Reims ACC is visualized with a purpose of confirming that transitions between active sectors respect all operational constraints. This Figure confirms that sector groupings/degroupings are done in the smooth manner without sudden and significant changes in the number of active sectors. It also shows solution stability, where sectors are kept active to maximum extent possible while all constraints are still satisfied, and objective function is not significantly penalized. The stability of the solution is important since every change of active sectors require a certain adaptation period for air traffic controllers to get used to the new working environment and lead to excessive workload that should be compensated with the objective function gains.

Wraping up, results presented in this section clearly confirm that ASP algorithms in the current ConOps (static sectorisation) are doing what was expected. It optimizes the number of active sectors, while respecting sector capacity constraints. At the same time, it provides smooth transition between active sectors and balances sector loads in the search for the fair solution.

3.1.2.2 Verification of the ASP in dynamic airspace configuration (DAC) mode (future ConOps)

The verification of the ASP in the DAC mode, due to difficulties mentioned in Section 4.2.2 was done for the ASP module solely and the simplified user case scenario that consider only Reims ACC. Using Test 4 traffic data, complexity metric for each of the 21 elementary sectors for 48 periods (24h separated by 30 minutes) during a day where computed (see Table 3-3). Figure 3-12 shows neighbouring graph between elementary sectors of the Reims ACC that was manually extracted. Based on those input data and empirically selected capacity value in terms of allowed sector complexity, ASP algorithm was computing grouping of the elementary sector for each period in the search for best sector grouping.



	P1	P2	P3	P4	P5	P6	...
S0	148.49949	119.258747	32.8967355	141.339715	140.781794	199.417574	...
S1	9.56976121	18.9546134	35.7505792	5.25262508	0.0079974	24.9736176	...
S2	26.8336815	30.0148768	7.51291015	5.37253619	3.50556374	4.78899757	...
S3	44.4766991	32.2681105	9.07366543	30.9239473	0.79878418	10.3514993	...
S4	44.3715014	62.666541	10.22741	16.84368	50.8121414	146.200177	...
S5	100.710744	41.3701436	35.043769	99.3562123	123.531065	119.225876	...
S6	5.82539971	62.3849151	11.095476	75.0403309	15.1470093	104.717289	...
S7	103.336124	152.69532	50.7699877	7.71100025	15.7270646	51.6300866	...
S8	112.452442	114.032771	47.5001847	49.1032019	67.8889422	21.7318285	...
S9	77.045419	120.779776	44.9294286	79.5875653	182.826799	128.074929	...
S10	320.109994	394.697123	100.973261	252.899637	601.587796	150.725059	...
S11	108.415513	110.649464	30.9171186	136.593	17.8476937	24.2922198	...
S12	15.0304618	9.31637234	21.9930712	5.33463935	23.9286692	8.33326742	...
S13	71.1289447	62.5917832	27.9870996	136.292694	130.490196	178.005429	...
S14	134.660522	64.8967813	99.8767554	153.868339	69.1173935	91.7096815	...
S15	237.418894	324.775771	119.024682	192.170455	91.277894	73.4756181	...
S16	39.9369575	88.5769416	59.0720628	46.3942442	90.3914532	70.7436977	...
S17	179.916903	303.382168	190.426295	192.279692	237.983057	83.9677672	...
S18	87.4984653	20.9813338	2.31776572	112.080655	31.6491064	5.60197947	...
S19	64.6407697	44.3229826	22.6352295	10.0447045	10.6566663	18.3099851	...
S20	54.7646129	210.529526	162.720919	67.7569501	2.85455031	207.682198	...

Table 3-3. Example of elementary sector complexity for Test4

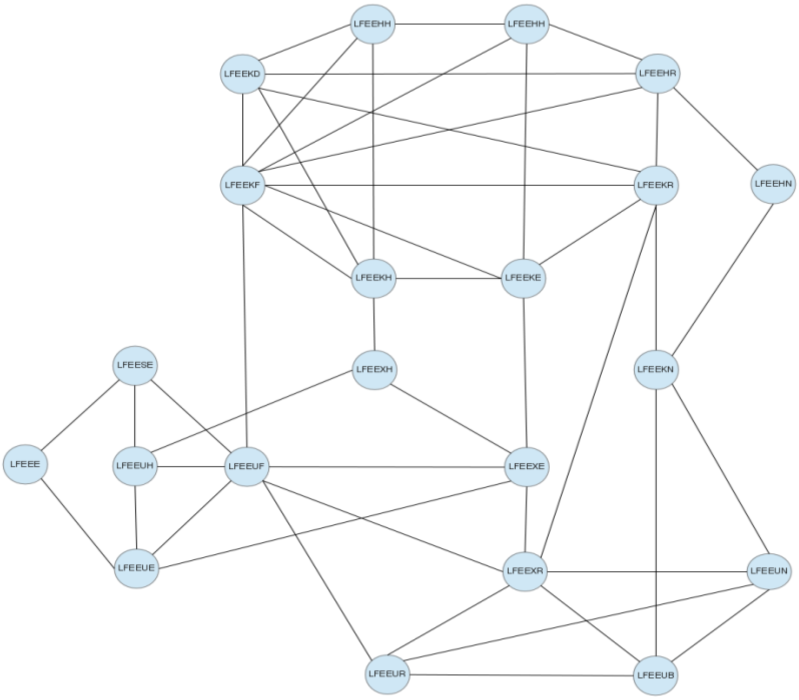


Figure 3-12. Neighbouring graph for the Reims ACC

Figure 3-13 visualize in 2D active sectors in the upper airspace of the Reims ACC for the three simulation periods: 13h, 14h and 15h, as a result of the ASP. The Figure show change of the sector grouping due to the change of the traffic.

The number of active sectors and traffic volume in the Reims ACC, in term of occupancy, are shown in Figure 3-14. This Figure shows good correlation of the number of active sector with the traffic demand, with number of active sector increasing/decreasing with traffic demand.

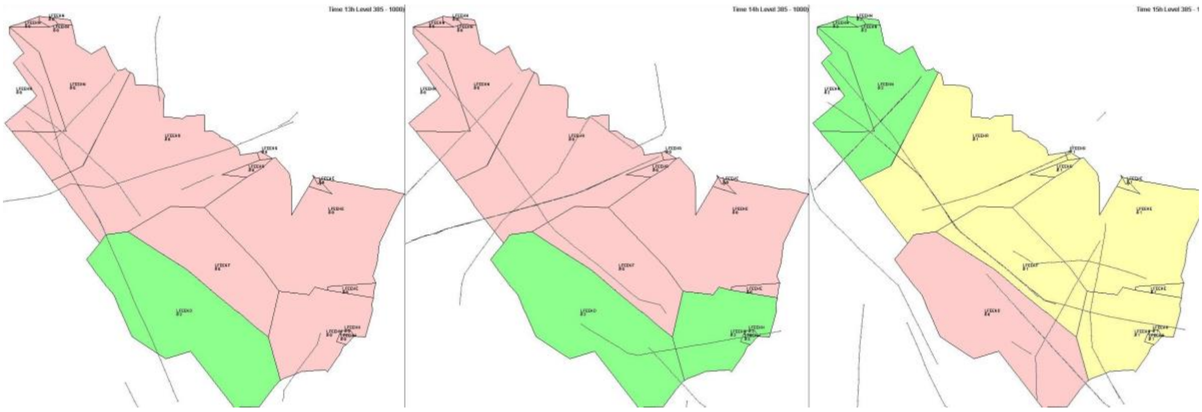


Figure 3-13. Example of the active sectors in the upper airspace for the 3 periods of time considered

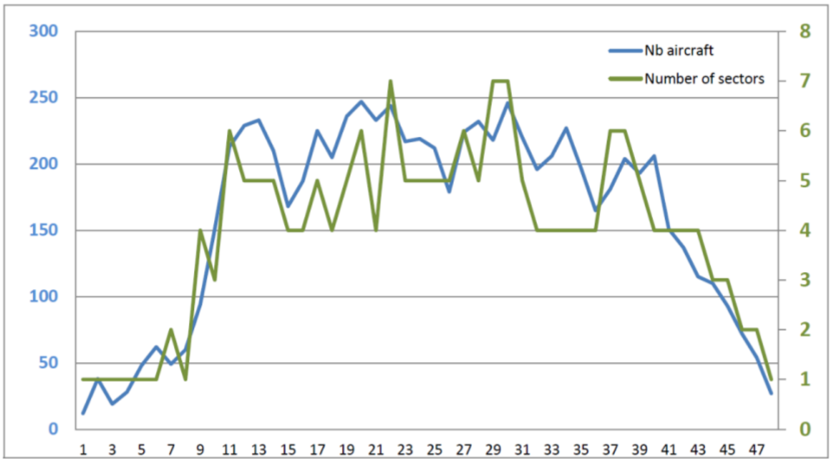


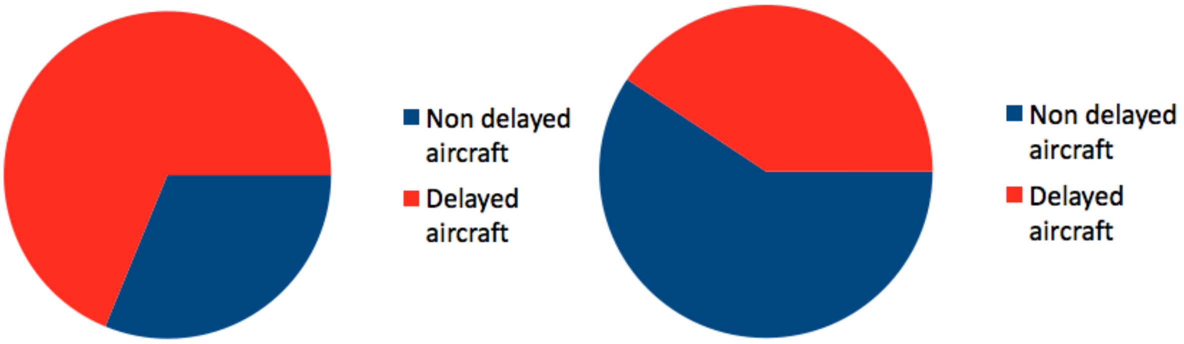
Figure 3-14. Distribution of active sector and traffic demand in the Reims ACC

The tests presented above verify that the ASP algorithm in the DAC mode is doing what is supposed to do, finding minimal necessary number of sector needed to service the traffic demand. Yet, the ASP was not able to output the results in the format necessary for the other modules (see section 4.2.2 for details). As a consequence, Test 4 was discarded and not fully achieved and the proper integration of the DAC mode is proposed as a further research topic that needs more detailed analysis that could not be devoted given the timeframe of the APACHE project.

3.1.3 Traffic and capacity planner (TCP) component

Once the ASP has generated an optimum sectorisation, trying to better allocate airspace capacity, the TCP is responsible to regulate the demand, avoiding to exceed the maximum capacity in any sector. The illustrative results shown here correspond to the 24h test benchmark described above, but focusing only in the French airspace. Both TCP modes of operation have been tested (current CASA algorithm and advanced DCB).

A comparison between both modes (Test Cases 1 and 3) is shown in Figure 3-15 for a regulation of 2 hours of high traffic demand. As expected, the number of delayed aircraft is higher with CASA than with ADCB, since the later allows also for (optimal) re-routeing and level cappings.



(a) Test Case 1: CASA (b) Test Case 3: ADCCB

Figure 3-15. Delays after regulating the demand for Test Cases 1 and 3

3.1.3.1 Verification of the TCP in CASA mode (current ConOps)

With the CASA algorithm, we tested 2 different parameters of execution. The first parameter is the threshold of demand overload allowed (i.e. the percentage of demand above the declared capacity for all sectors).

Figure 3-16 shows the traffic demand, as obtained from Eurocontrol's DDR2, for an example sector in the French airspace (FRRESTU). In DDR2 it is also reported that around 8h00 an ATFM regulation took place (in red in the figure) in that sector. The nominal capacity declared is the green horizontal line in the Figure, while the dark blue bars represent the initial traffic and light blue bars the regulated traffic. As it can be observed in Figure 3-16 not always a demand above the nominal capacity triggers a regulation. Moreover, when the regulation is declared the new capacity (short red line in the Figure) is above the original nominal capacity.

This visual example higlights something that is widely known: in real operations, each hotspot (i.e. sector with demand above nominal delcared capacity) is carefully analysed by the corresponding flight management position (FMP), which have different strategies and criteria to finally decide whether a regulation should be applied or not, if a new capacity (higher than the nominal one) should be declared, if some overload (i.e. demand above nominal capacity) is allowed for certain sectors in certain periods of time, etc. This behaviour strongly relies on (expert) human intervention and decision making (the staff working at the FMP) and is very difficult to model it by a computer program such as the APACHE TCP.

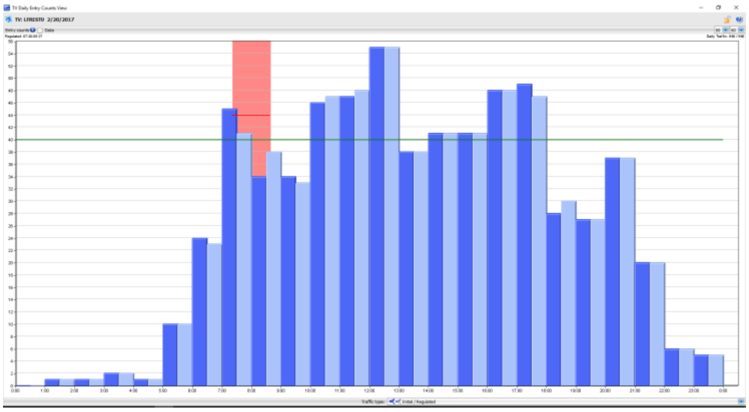


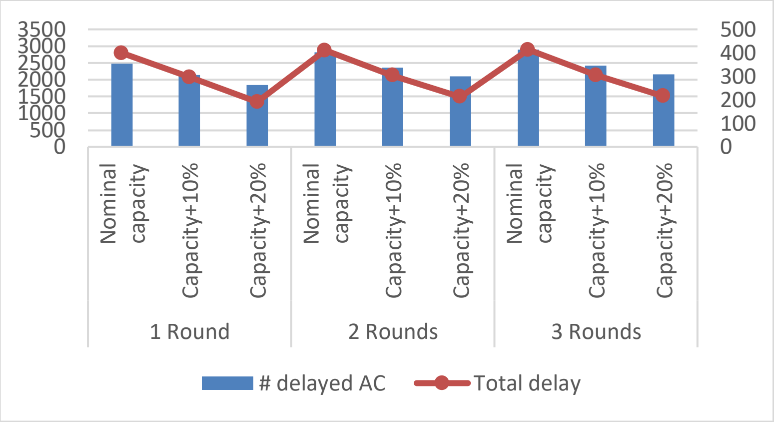
Figure 3-16. Example of regulations applied currently in France for sector LFRRESTU (capacity is the fixed green line, dark blue is the initial traffic, light blue is the regulated traffic and regulation is shown in red)

Since it is out of the scope of APACHE to accurately model the behaviour of each FMP, and therefore, these ad-hoc actions are not considered in the TCP. Instead, two parameters have been tested. First, allowing different **sector overloads**: 10% and 20%. Secondly, the **number of iterations** of the CASA algorithm: after applying CASA for the first time (taking the most penalisng delay for those aircraft affected by more than one regulation), new sectors that were not regulated before can become overloaded. CASA can solve this by iterating over and over, until no more sectors are overloaded.

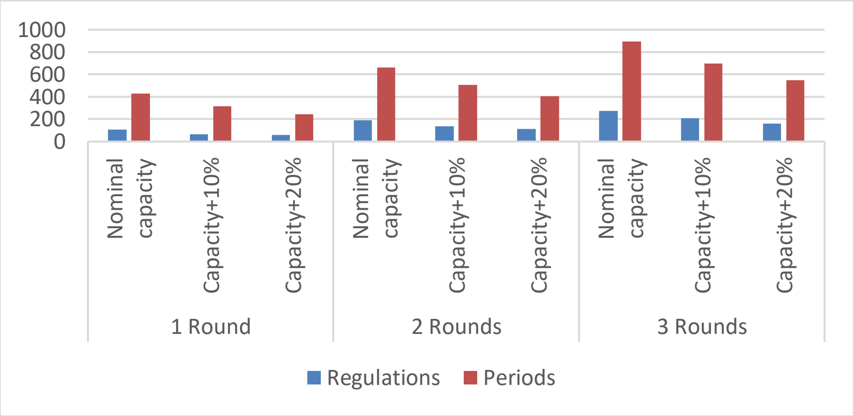
Results are shown in Figure 3-17. As the overload threshold increases, fewer regulations appear and thus less delay is necessary (Figure 3-17a). The second and third interation of the CASA execution also increments the applied delays, but not very significatively.

A similar behaviour can be seen in Figure 3-17b, which shows the number of sectors regulated and the duration of such regulations in periods of 20 minutes. Regulations and duration of regulations are decreasing when additional capacity is granted. When more iterations are executed, more regulations are needed. Sectors regulated again in another iteration are accounted twice. Although the algorithm needs more time to go through the new regulations, the actual delays are not much affected.

For all aforementioned reasons, it was **finally decided to fix the values of the two parameters** of the CASA algorithm (when using it in WP5 for the APACHE validation exercises) to: **20% of allowed overload** (with respect to nominal capacity declared) for all sectors; and **one round** of CASA algorithm.



(a) Total delay

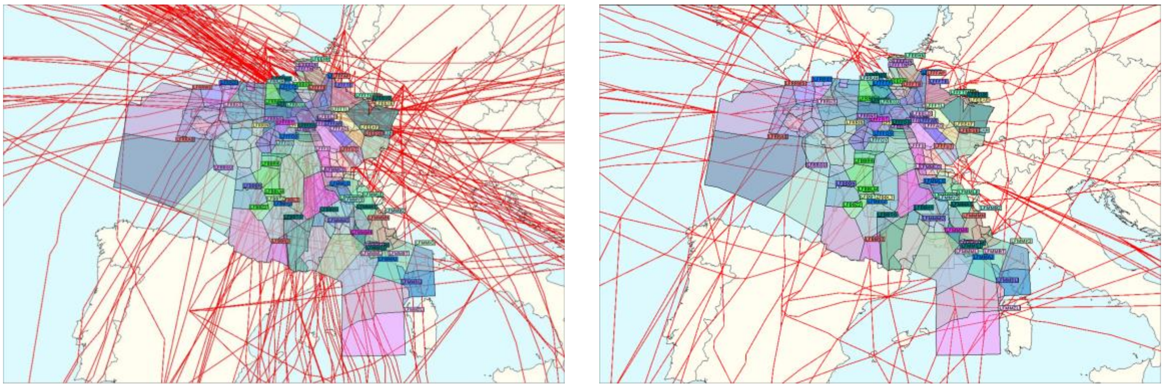


(a) Number of regulations

Figure 3-17. Results using CASA with different sector overload allowances and algorithm rounds

3.1.3.2 Verification of the TCP in ADCB mode (future ConOps)

The simulation scenario is focused on the French airspace with 24 hours' traffic scheduled to traverse this area. The unit time slot in the experiments is set to be 1 min, while the time scale for capacity counting is 20 min (i.e., the number of flight entries in a sector per 20 min). As explained in Table 3-1, the ADCB mode was tested against two different sets of traffic: assuming a full free route scenario (Test 2) and considering a structured route network when optimising the trajectories (Test 3). Figure 3-18 shows the initial demand for both Test Cases and the elementary sectors of the French airspace used for these Test Cases.



(a) Test Case 2 (Free Route)

(b) Test Case 3 (Structured routes)

Figure 3-18. Initial demand (flight over France) and elementary sectors of the French airspace

The sample data involve **6,593 flights** in total that were scheduled to fly through the area between 0h and 24h in Feb 20th 2017. However, there appear some cases (which is not uncommon) that a trajectory only forms a small part of intersection with a sector. Due to the operational limits, such as communications, the responsibility associated with this short flight path might not need to be transferred to ATCos handling this sector. Hence, the temporal intersection should not be counted as an independent flight entry. In this study, **60 seconds** is regarded as the minimal time spent in a sector.

Accordingly, after removing those initial trajectories with all their sector intersections less than 60 seconds, there are **6,387** and **6,255** flights left for respectively Test 2 and Test 3, which in turn will be subject to further regulations. On the other hand, the total number of elementary sectors is 164 for that day, which are merged into 224 different collapsed sectors through the 72 time periods during 24 hours (i.e., each time period lasts for 20 min).

The **first stage** of the ADCB algorithm is the hotspot detection, i.e. those sectors (and time periods) where the forecast traffic demand is above the nominal capacity. For Test 2, 115 hotspots were detected, being 86 for Test 3. The regulated flights (i.e. those flights crossing the hotspots) were 1,813 and 1,464 respectively.

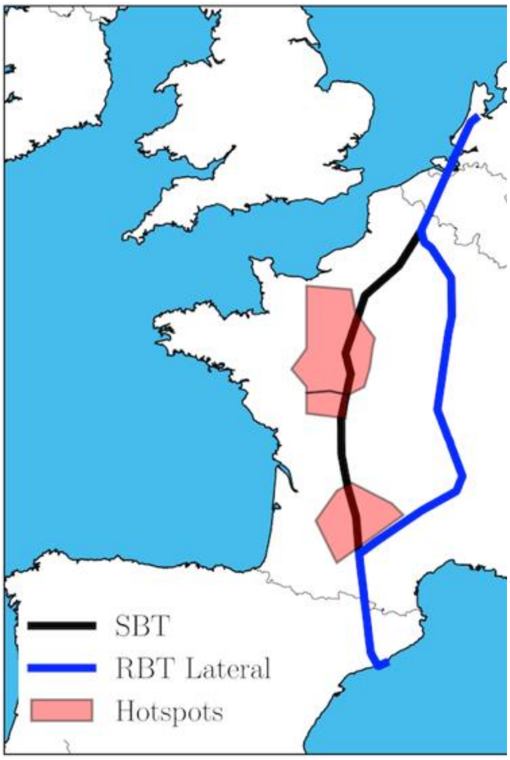
The **second stage** is to request to the APACHE TP (emulating the AUs in real operations) alternative trajectories for these regulated flights in order to avoid these hotspots. When operationally possible[11],

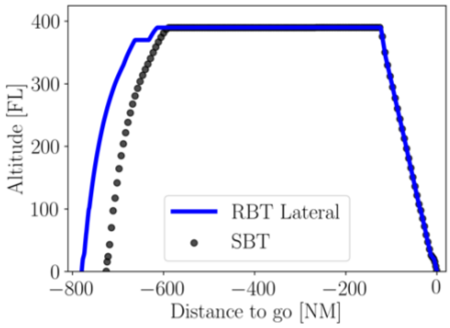
---

[11] It should be noted that lateral or vertical re-routings are not always possible. For example, if the hotspot contains the destination airport, the sector cannot be avoided laterally. Moreover, it is not always operationally possible to avoid a sector in the vertical domain due to aircraft performance limitations. For this reason, each regulated flight will have from one, up to

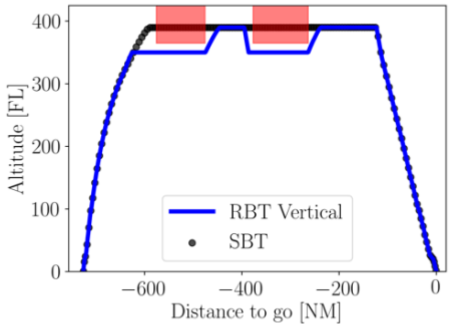
the APACHE TP returned two alternative trajectories: a trajectory avoiding the hotspots laterally (re-routing); and a trajectory avoiding the hotspot vertically. Table 3-4 summarises these results for the two Test Cases, while Figure 3-19 gives an example of a lateral avoidance of two hotspots (showing also the new resulting vertical profile) and an example of the vertical avoidance of the same hotspots. These new (alternative) trajectories are computed by the APACHE TP and are the best trajectories (i.e. optimal trajectories) such that avoid the two hotspots and given the en-route structure of available airways.

The third and last stage is the ADCB algorithm that selects the best combination of trajectories and/or delays in such a way that a global (system-wide) cost function is minimised. This optimisation process is described in Deliverable D3.2. The total number of trajectory options that will be considered by this algorithm accounts for the initial flight plan (that can be eventually delayed), plus the lateral and vertical trajectory alternatives for those flights affected by a hotspot[12].





(b) New vertical profile resulting from lateral avoidance



(a) Lateral avoidance (re-routing) (b) Vertical avoidance (level capping)

Figure 3-19. Example of alternative trajectories (to avoid hotspots) provided by the TP to the TCP for Test 3 (structured routes and FL allocation/orientation schemes)

---

3 different options to solve the ADCB problem: only delay (keeping the original trajectory) and then eventually lateral and/or vertical alternatives.

[12] It should be noted that regulations on flights affected by an initial hotspot (delays, re-routings or changes in cruise altitude) can eventually create other hotspots in the network (i.e. saturate sectors that were not initially congested). Thus, aiming at fining the system-wide optimum and keeping demand below capacity in ALL network sectors, the ADCB algorithm may delay aircraft that were not crossing one of the initial hotspots. This is why delaying the original trajectory is always an option for the ADCB algorithm (6,387 and 6,255 trajectory options for Test 2 and 3, respectively).

	Demand (flights)	Lateral trajectory alternatives	Vertical trajectory alternatives	Total number of trajectory options
Test 2 (full free route)	6,387	1,628	1,727	9,742
Test 3 (structured routes)	6,255	1,305	1,379	8,939

Table 3-4. Summary of trajectory options for both Test Cases

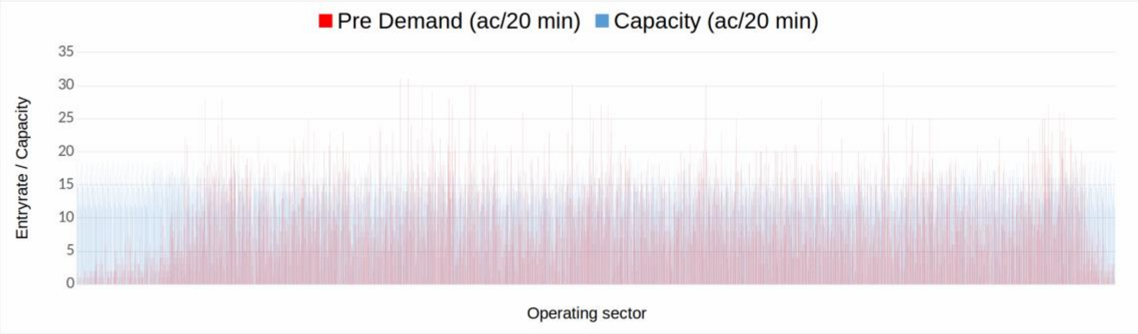
For the two Test Cases presented here, some assumptions have been made when running the optimisation algorithm:

- the costs of time adjustments have been set linear, and apply the same across all the flights (e.g., 15 euro/min for ground holding and 20 euro/min for air holding including the standard airborne holding and linear holding);
- the time upper bound for performing linear holding has been set to 20% of the segment flight time, based on the statistical average value derived from former work (Xu et al. 2017), and for delay recovery this bound is set to be 10%, both of which are rounded to the greatest integer that is less than or equal to;
- the cost of delay recovery has been set to -5 euro/min, meaning that all the flights would favour to increasing certain speed (burning some extra fuel) to recover part of their previously experienced delays (if any). It must be noted that the delay recovery is only allowed when a flight is assigned with some delays at its trajectory forepart, such as ground holding at the origin airport;
- the price of fuel is assumed to 0.4 euro/litre, i.e., about 0.5 euro/kg; and
- the route charges have been calculated based on the absolute distance flown inside an area, rather than the great circle distance between the entry and exit positions, which is the current policy.

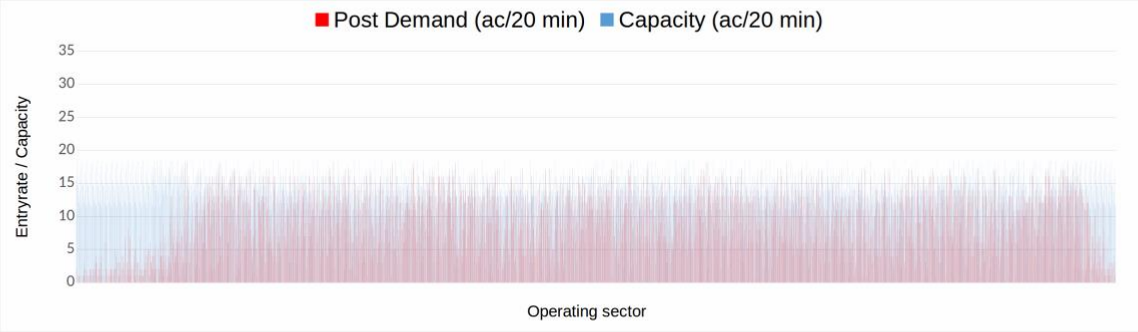
Figures. 3-20a and 3-21a present firstly the initial (i.e., pre-regulation) demand versus capacity for each considered operating sectors. Indeed, large numbers of capacity overloads can be found, while in some cases it could be as high as twice the capacity value that the sector can provide. Moreover, the situation tends to be even worse for the free route case. After the regulation, as expected, it can be seen in Figures. 3-20b and 3-21b that all the exceeded demands have been balanced below the respective operating sectors' capacities.

To further understand the balance between demand and capacity, their ratios are sorted (based on pre-regulation) and presented in Figure 3-21. The curves representing pre-regulation are steeper with some parts growing higher than 1, meaning that for those operating sectors the flight entries are higher than their capacities. Conversely, the curves turn to be level and average with respect to the post-regulation cases, which means that more airspace capacities are well utilized.

The dimensions of the problem, for each of the two Test Cases, are summarized in Table 3-5. In the numerical experiments, GAMS v.24.2 software suite has been used as the modelling tool and Gurobi v.5.6 optimizer has been used as the solver. The numerical experiments have been run on a 64-bit Intel i7-4790 @ 3.60 GHz quad core CPU computer with 16 GB of RAM memory and Linux OS.

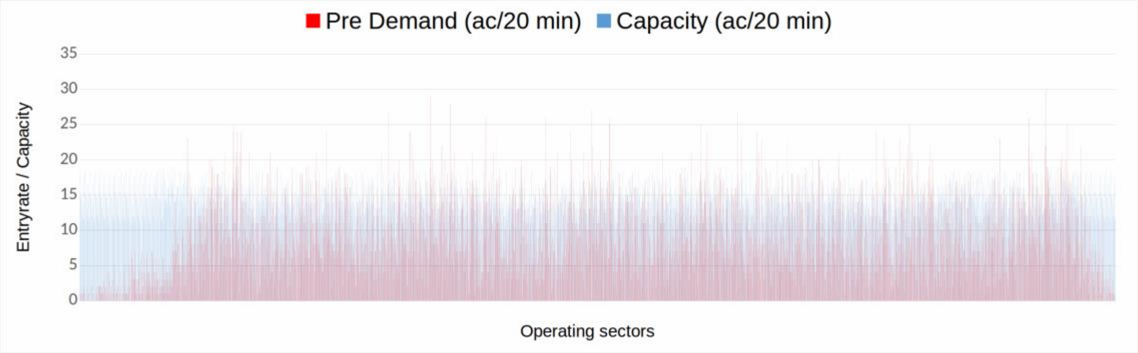


(a) Pre-regulation flight entries and operating sector capacities

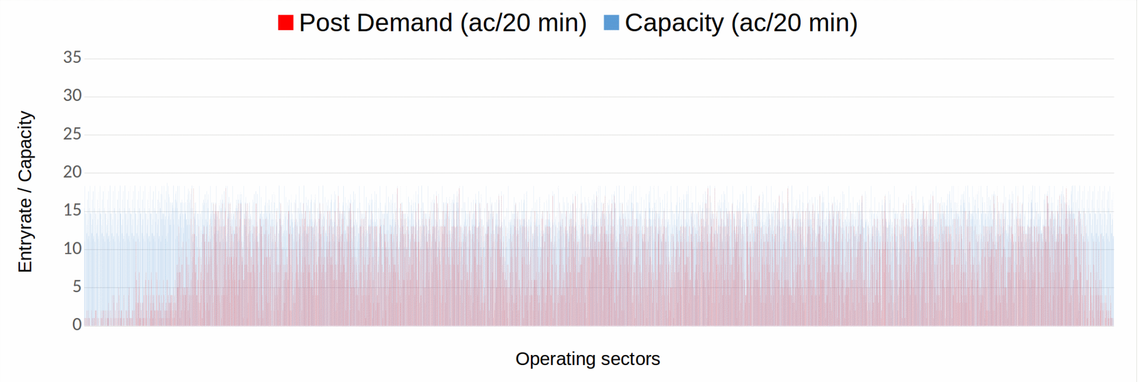


(b) Post-regulation flight entries and operating sector capacities

Figure 3-20. Overall traffic demand vs. airspace capacity for Test Case 2 (Free route)

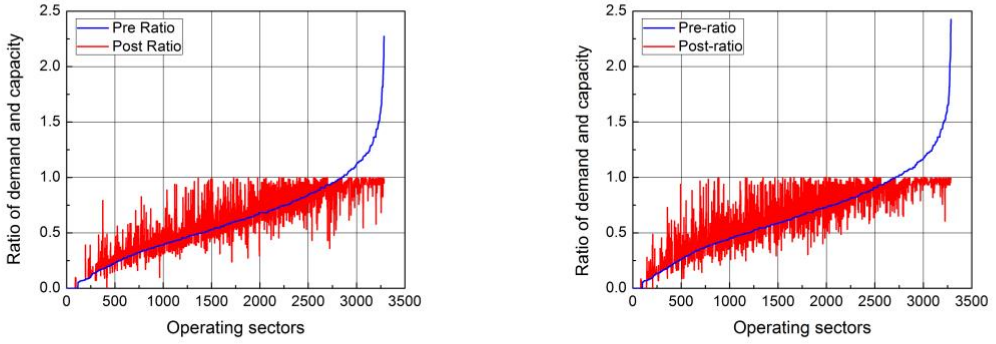


(a) Pre-regulation flight entries and operating sector capacities



(b) Post-regulation flight entries and operating sector capacities

Figure 3-21. Overall traffic demand vs. airspace capacity for Test Case 3 (Structured routes)



(a) Test Case 2 (Free route) (b) Test Case 3 (structured routes)

Figure 3-22 Demand/capacity ratio for pre- and post-regulation (sorted by pre-regulation ratio)

	Test Case 2	Test Case 3		Test Case 2	Test Case 3
Variables	5546029	4822740	Solution time (min)	240	150
Equations	13045946	11307028	Objective value	240768	129027
Non-zero elements	28646435	27543864	Relative gap	0,10%	0,05%
Generation time (min)	30	20			

Table 3-5. Problem size and computational time for the cases of study

Cases	Total delayed flights (a/c)	Total delay (min)
Test Case 2 (GH mode)	1,798	207,506
Test Case 3 (GH mode)	1,840	219,862

Table 3-6. Benchmark of assigned delays and affected flights in "only with Ground Holding" mode

Table 3-6 presents a set of benchmark results, where **GH mode** means that all the possible measures in the ADCB will be disabled, except for ground holding: similar to CASA algorithm, but minimising the total delay in the objective function and not applying a ration by schedule policy as done in the CASA algorithm.

The full version of the ADCB takes into account ground holding (GH), air holding (AH), linear holding (LH), and delay recovery (DR). See (Xu and Prats, 2017) for details. Moreover, the ADCB imposes arrival delay (AD) at the destination airport (or congested sector) instead of imposing departure delay at the origin airport as done nowadays for ATFM regulations.

Detailed results of trajectory options and timeline adjustments can be appreciated from Table 3-7 to Table 3-10. The most promising result would be that the total (arrival) delay is reduced respectively to 5,836 min for Test Case 2 (see Table 3-7) and 4,691 min for Test Case 3 (see Table 3-8).

If comparing the total number of regulated flights, the difference between the ADCB GH mode and full mode is relatively small. For the GH mode, the only available measure is ground holding, and the flights captured to execute it are 1,798 and 1,840 respectively (see Table 3-6). For the full version, the regulated flights (i.e., performing any of the available measures) are at least 2,140 (i.e., 6,387 - 4,247) for Test Case 2 and 1,768 (i.e., 6,255 - 4,487) for Test Case 3. This is shown in Tables 3-9 and 3-10, respectively.

Options	Initial trajectory (a/c)		Lateral alternative (a/c)		Vertical alternative (a/c)		Total (a/c)	
	5434		376		577		6387	
	Flights (a/c)	Time (min)	Flight (a/c)	Time (min)	Flight (a/c)	Time (min)	Flight (a/c)	Time (min)
GH	1173	6362	89	278	149	467	1411	7107
Non-GH	4261	-	287	-	428	-	4976	-
AH	54	308	4	9	7	10	65	327
Non-AH	5380	-	372	-	570	-	6322	-
LH	87	189	6	14	8	13	101	216
Non-LH	5347	-	370	-	569	-	6286	-
DR	837	-1516	69	-140	107	-158	1013	-1814
Non-DR	4597	-	307	-	470	-	5374	-
AD	890	5343	54	161	110	332	1054	5836
Non-AD	4544	-	322	-	467	-	5333	-

Table 3-7. Summary of trajectory options and timeline adjustment for Test Case 2 (free route)

Mix	Initial trajectory			Lat. alternative			Vert. alternative			Total		
	Flight (a/c)	GH (min)	AD (min)	Flight (a/c)	GH (min)	AD (min)	Flight (a/c)	GH (min)	AD (min)	Flight (a/c)	GH (min)	AD (min)
GH + AD	876	5882	5260	51	201	148	106	408	327	1033	6491	5735
GH + Non-AD	297	480	-	38	77	-	43	59	-	378	616	-
Non-GH + AD	14	-	83	3	-	13	4	-	5	21	-	101
Non-GH + Non-AD	4247	-	-	284	-	-	424	-	-	4955	-	-

Table 3-8. Examples of mixed timeline adjustments for Test Case 2 (Free route)

Options	Initial trajectory (a/c)		Lateral alternative (a/c)		Vertical alternative (a/c)		Total (a/c)	
	5546		388		321		6255	
	Flight (a/c)	Time (min)	Flight (a/c)	Time (min)	Flight (a/c)	Time (min)	Flight (a/c)	Time (min)
GH	1052	5269	58	253	72	357	1182	5879
Non-GH	4494	-	330	-	249	-	5073	-
AH	34	249	1	1	4	34	39	284
Non-AH	5512	-	387	-	317	-	6216	-
LH	58	148	1	3	8	18	67	169
Non-LH	5488	-	387	-	313	-	6188	-
DR	765	-1443	52	-106	55	-92	872	-1641
Non-DR	4781	-	336	-	266	-	5383	-
AD	751	4223	38	151	59	317	848	4691

Options	Initial trajectory (a/c)		Lateral alternative (a/c)		Vertical alternative (a/c)		Total (a/c)	
	5546		388		321		6255	
	Flight (a/c)	Time (min)	Flight (a/c)	Time (min)	Flight (a/c)	Time (min)	Flight (a/c)	Time (min)
Non-AD	4795	-	350	-	262	-	5407	-

Table 3-9. Summary of trajectory options and timeline adjustment for Test Case 3 (structured routes)

Mix	Initial trajectory			Lat. alternative			Vert. alternative			Total		
	Flight (a/c)	GH (min)	AD (min)	Flight (a/c)	GH (min)	AD (min)	Flight (a/c)	GH (min)	AD (min)	Flight (a/c)	GH (min)	AD (min)
GH + AD	744	4678	4187	38	214	151	57	330	314	839	5222	4652
GH + Non-AD	308	591	-	20	39	-	15	27	-	343	657	-
Non-GH + AD	7	-	36	0	-	0	2	-	3	9	-	39
Non-GH + Non-AD	4487	-	-	330	-	-	247	-	-	5064	-	-

Table 3-10. Examples of mixed timeline adjustments for Test Case 3 (Structured Routes)

3.1.4 Risk assessment (RA) component

The Risk Assessment (RA) component is intended for simulation of air traffic consisting of optimal flights trajectories (output of the Trajectory Planner and/or the Traffic and Capacity Planner components) through a given airspace sectorisation (output from Airspace Planner component) with the aim to assess safety performances and to provide outputs in form of Safety KPIs. The RA component is consisting of three modules.

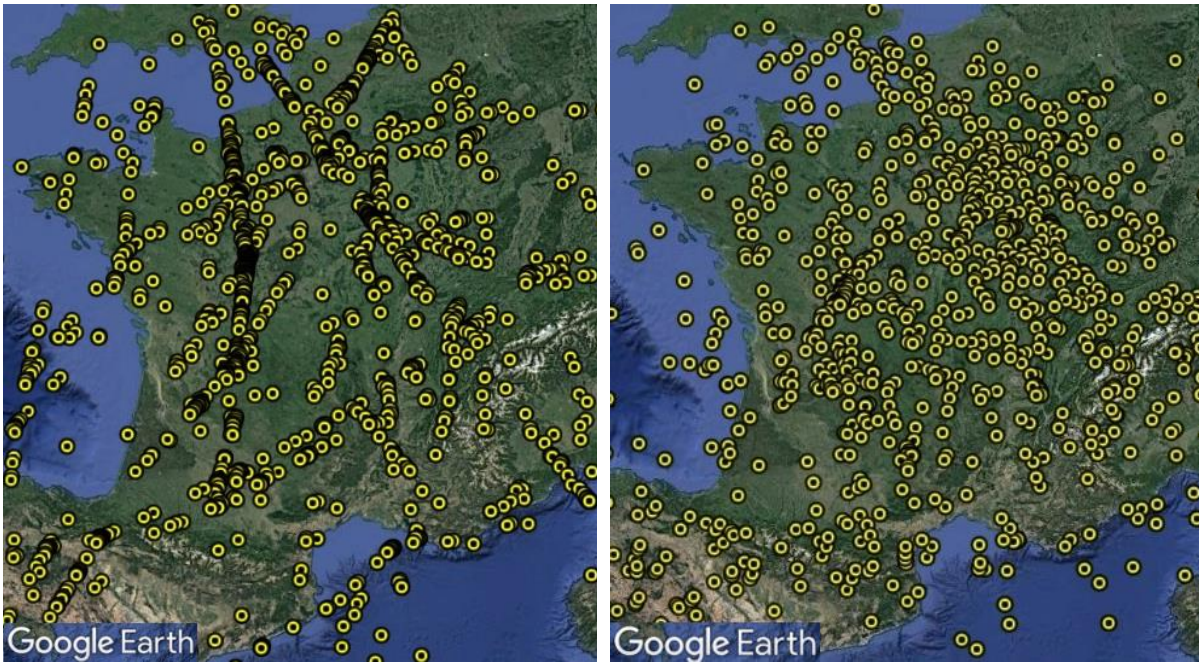
- Separation violation detection module;
- TCAS activation module; and
- risk of conflict/accident assessment module.

The RA component is based on the assumption that conflict between pair of aircraft exists when either horizontal and/or vertical separation minima are violated. The Separation violation detection module compares actual separation of aircraft (both in horizontal and vertical plane) with given separation minima in order to detect potential conflict. Once conflict is detected this module counts them (see SAF-4 performance indicator) and then for each conflict calculates severity (SAF-5 indicator) and duration (SAF-6) of conflict situation in the observed airspace under given circumstances. If the situation worsens the TCAS activation module is activated. It counts Traffic Alerts (SAF-1) and Resolution Advisories (SAF-2) warnings and based on them number of NMACs (SAF-3). All previous Safety (SAF) indicators are defined in (APACHE, 2018).

The risk of conflict/accident assessment module is based on calculation of “elementary risk” which is defined as the area between the surface limited by the minimum separation line and the function representing the change of aircraft separation. The risk of conflict/accident (SAF-7) is then defined as the ratio between the “elementary risk” and the observed period of time. Apart from the risk between specific aircraft pairs, an assessment of the total risk in a given sector is also considered (Netjasov, 2012).

A risk assessment was done for the selected day of study and for both sets of trajectories generated by the TP but filtering for those flights crossing only the French airspace. Table 3-11 shows the results of all safety PIs, as computed by the RA module, for all Test Cases. The minimum separation values (for SAF-4) were set to 5NM in the horizontal plan and 1000 ft in vertical. Moreover, the simulation time increment was set to 10s. As expected, those indicators are lower for the full free route scenario since potential trajectory crossings are more geographically spread (Netjasov & Crnogorac, 2018).

Fig. 3-23 shows the geographical location of the closest points of approach (CPA) that were below 5NM in the horizontal plane or 1000ft in the vertical plane for the example of study (SAF-4 indicator). One should keep in mind that CPAs shown are aggregated for 24h, which means that each dot represent a conflict point between different pair of aircraft, at different altitudes and in different time during the day. Also note that even if the test flight set corresponded to flights crossing the French airspace during 24h, CPAs could be located outside this airspace, since the full trajectory was taken into account (Netjasov & Crnogorac, 2018).



(a) Test1 (current route structure) (b) Test2 (full free route)

Figure 3-23 Location of conflicts (CPA below 5NM horizontal or 1000 ft vertical) (Netjasov & Crnogorac, 2018).

PI	Test Case 1	Test Case 2	Test Case 3
SAF-1	361	197	420
SAF-2	146	3	93
SAF-3	107	1	73
SAF-4	1816	829	1400
SAF-5 (average)	0.523 ± 0.297	0.470 ± 0.271	0.560 ± 0.302

PI	Test Case 1	Test Case 2	Test Case 3
SAF-6 (average)	352.22 ± 754.45	112.36 ± 299.21	303.14 ± 785.90
SAF-7	$7.3 \cdot 10^{-3}$	$3.0 \cdot 10^{-3}$	$6.0 \cdot 10^{-3}$

Table 3-11. SAF PIs for the test cases (Netjasov & Crnogorac, 2018).

3.2 Validation results for the APACHE System components

This section contains the different exercices performed to validate each component of the APACHE System. As explained before, each APACHE System component has been validated independently, meaning that the setup and configuration of the component and input data used might be different for each validation. This section details all these validation tests.

3.2.1 Trajectory planner (TP) component

Two different validation exercises have been done:

- comparison with trajectories generated with the **Airbus performance engineering programs** (PEP); and
- comparison with trajectories generated by the **AURORA SESAR Exploratory Research Project**.

PEP is an application designed to provide flight performance engineers with the necessary tools to handle the performance aspects of flight preparation, and also to analyse aircraft performance after the flight. The Airbus PEP comprises several modules. The flight planning module (**FLIP**) allows to produce fuel predictions for a given flight under simplified meteorological conditions (i.e. constant wind), accounting also for airline cost policies and aircraft performance capabilities. Trajectories obtained from FLIP assist dispatchers in determining the optimum fuel quantity to be carried, as long as optimal cruise level(s) and speeds, as a function of the payload, the ground distance from origin to destination and the Cost Index (CI). In addition, these trajectories are computed using performance data from the manufacturer and optimisation algorithms similar to those installed in the Flight Management Systems (FMS).

This first validation exercise consisted in comparing the vertical profile of some trajectories computed with the APACHE TP with those obtained with the Airbus PEP software suite for the same input parameters. The metrics for the comparison are the relative differences in flight time and fuel consumption figures, and the discrepancies in the optimal altitude and speeds profiles.

The validation was successful and all details are found in Appendix F (section F.1) of this Deliverable. This validation demonstrates that the trajectories computed by the APACHE TP are accurate in terms of altitude and speed profiles and also in terms of fuel consumption and flight time figures. Yet, the scope of this validation exercise only covered Airbus models.

The second validation exercise was done by comparing, for a same input test case, the outcomes of the APACHE TP (developed by UPC) with those obtained by the homologous tool used in the AURORA project (developed by Boeing Research and Technology Europe). Appendix F (section F.2) of this Deliverable contains details on this validation, which was done comparing a data set using 1500+ trajectories, covering several aircraft models, and focusing in comparing figure for fuel consumption; trip distance and time; cruise speeds; and cruise altitudes.

Validation was also successful, especially for the horizontal components of the trajectories, demonstrating that the route optimisation is the same for both TPs. Regarding the vertical profile, results show that both APACHE and AURORA TPs make use of the same weather data, validating their respective weather data processing and modelling functions. Yet, some differences in the altitude and speed profiles are observed, which, in turn, lead to discrepancies in the fuel consumption and flight time. Several factors have been identified as potential causes, mainly: different aircraft performance models (BADA 4.x vs. BADA 3.x), and different operational models and considerations when simulating aircraft climbs. Detailed results and explanations are given in Appendix F.

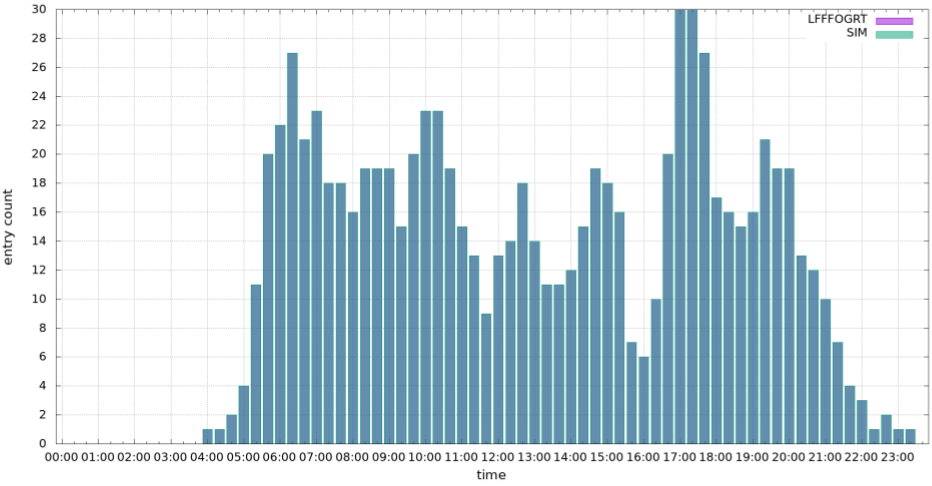
3.2.2 Airspace planner (ASP) component

Validation tests are performed with the purpose to confirm that the optimization of the number of active sectors done by the ASP, which has been previously verified (see Section 3.1.2), is done in a proper manner and that results reflect current operations. Several validation tests were designed and can be grouped in two categories: firstly, validating sector load computations; and secondly, validating the resulting sector opening scheme based on the given input traffic demand.

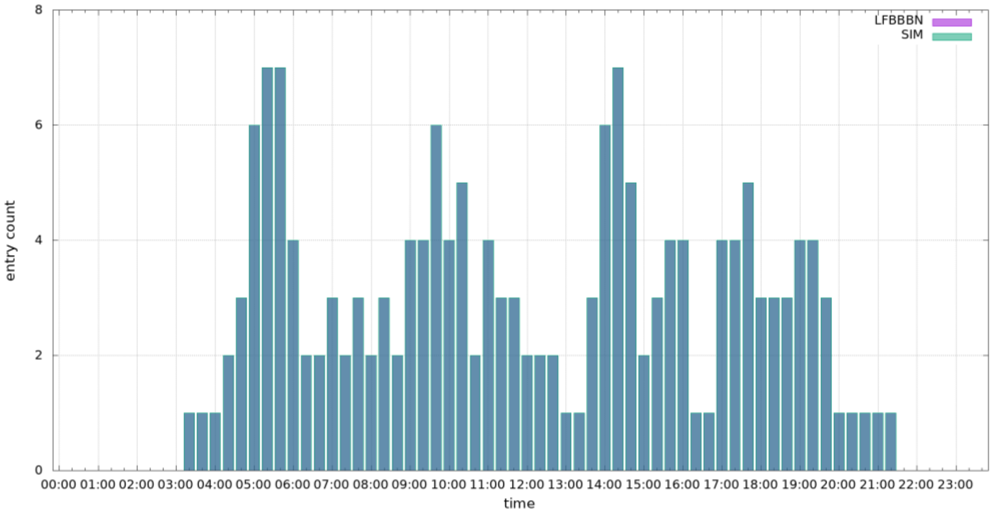
The first test performed considered sector load (entry count) computation. As a difference from verification tests (see Section 3.1.2), which were performed with the purpose of investigating the reasons for the high sector overloads in the upper airspace, validation tests presented here are designed to show that for the same traffic demand sector entry counts of the ASP algorithm are equal to real values of the traffic loads. Since there is no database available of historical sector loads, other validated models are used as a source to compute these baseline sector loads. As primary source, NEST Airspace load analysing tool is used, but validation also considered comparison with the APACHE TCP entry count calculation process.

The validation with NEST Airspace load tool included the following workflow. First, traffic demand in the so6 format is selected, representing either historical DDR traffic data or synthesised traffic data from the TP simulator. Multiple sources intended to secure general conclusion about ASP entry counts validation. In the next step, the so6 file is transformed in the t5 file using the NEST Airspace Traffic intersection tool. Finally, the t5 file is used independently by the ASP and NEST Airspace load tool to calculate sector loads, which are further compared and analysed (see Figure 3-24 for three example sectors).

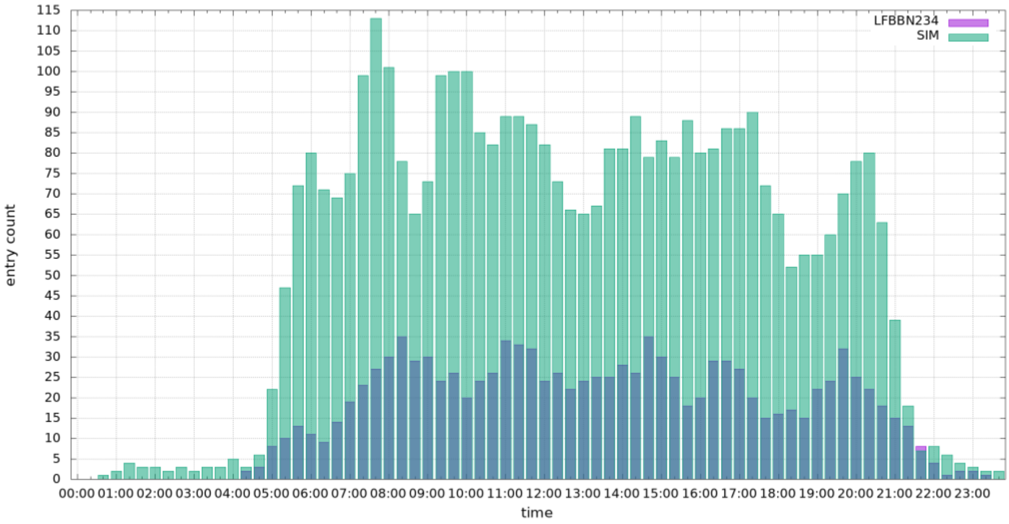
Regardless of the traffic data source, computed sector load values for both tools and all periods over the day showed perfect matching, all except one (see Figure 3-24c). Collapsed sector LFBBN234 drove attention due to huge difference in the computed sector load. Detailed analysis revealed that mismatching was due to wrong sector definition in the DDR database, where instead of grouping of elementary sectors N2, N3, and N4, collapsed sector N234 was defined as grouping of sectors L2, L3, L4, R2, R3, and R4.



(a) LFFOGRT sector load



(b) LFBBBN sector load



(c) LFBBN234 sector load

Figure 3-24. Example of ASP and NEST sector load comparison

Validation of the sector opening scheme, however, was not an easy task. Having access to the historical opening scheme data, the initial idea was to compare results of the ASP module with realized opening schemes for the same historical traffic data. This idea was supported with hypothesis that in the current system capacity is managed in a way to minimize traffic regulations. It is known, however, that DDR database does not contain reliable opening scheme information[13], and therefore this was disregarded as a source of realized data. As an alternative source of data, French national system CAUTRA was used to collect realized sector opening schemes. For the validation test, Feb 20th 2017 was selected, (the same day as for the verification tests), but instead of synthesised data, historical traffic data were taken from DDR database (regulated traffic - M2). Optimal opening scheme, as a result of the ASP algorithm, was then compared with collected scheme from the CAUTRA database (see Figure 3-25). Besides a similar tendency and similar peaks, this Figure shows very different results in the terms of number of active sectors between the ASP and the realized scheme. A detailed analysis has identified several reasons.

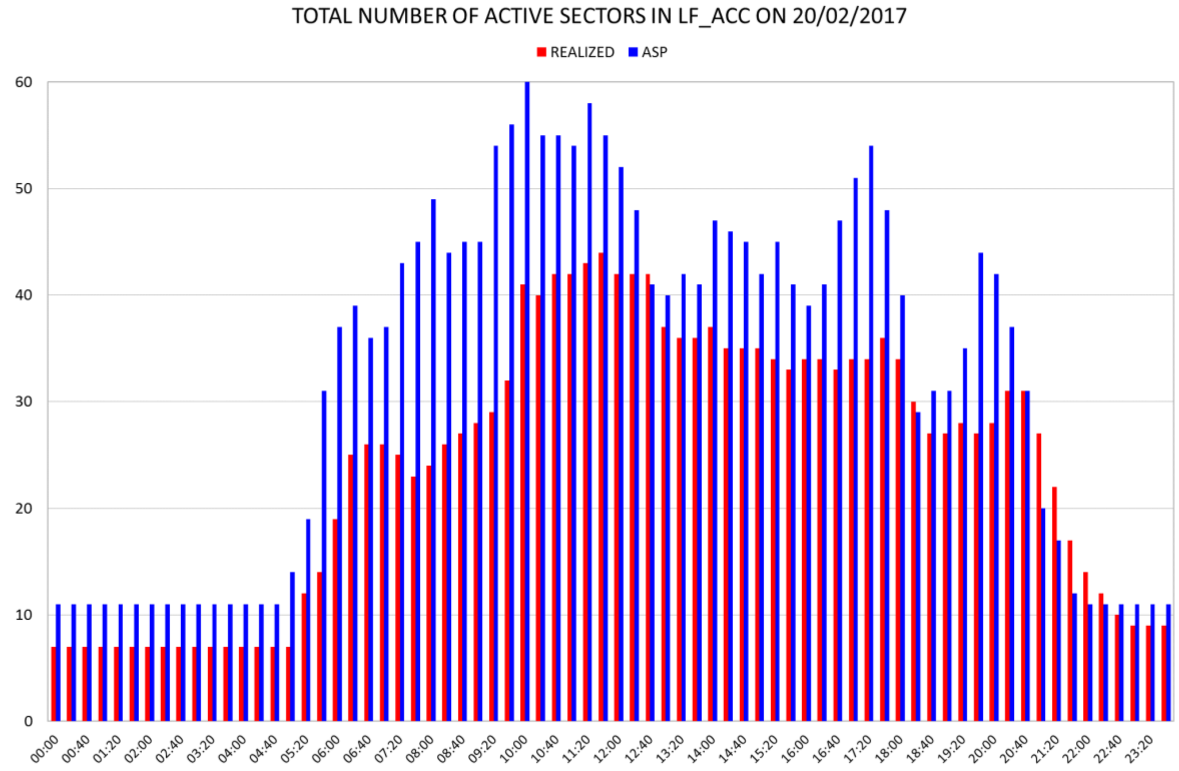


Figure 3-25. Comparison of realized and optimal number of active sectors in French airspace

First reason is the different repartition of the French ACCs in the airspace clusters: while AIRAC data, presented in the DDR database (and used by ASP) had 11 clusters in total, CAUTRA database had only 7 clusters. This difference is obvious during periods of the low traffic demand (early morning and late night) when minimal number of sector is activated.

---

[13] DDR contains only previsioned sector opening schemes, which are usually field automatically based on the period of the year, day of operations etc. Those schemes are revised during day of operations but rarely updated in the database.

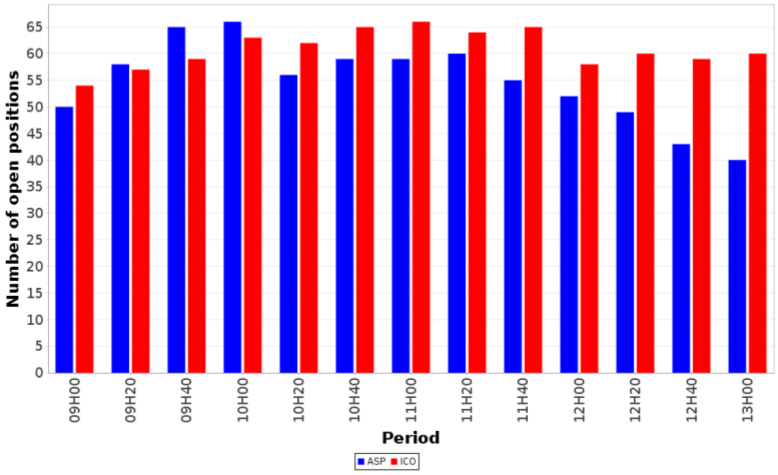


Figure 3-26. Distribution of the number of opened positions.

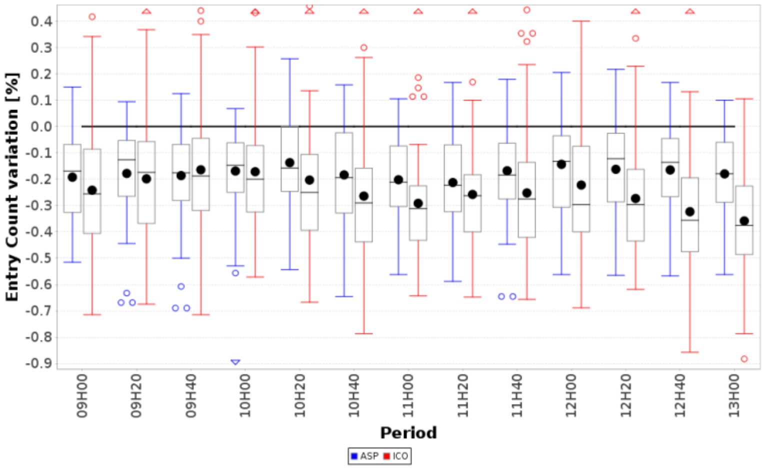


Figure 3-27. Distribution of the active sector load.

Secondly, a closer look at the airspace configurations used in the realized opening scheme revealed new configurations previously not defined and therefore not available to the ASP module. Those configurations were better adapted to the traffic demand, respecting sector capacity constrains with lower number of active sectors. They are manually selected based on FMP experience during day of operation. Just for illustration, in the Bordeaux ACC among 900 different airspace configurations used during 2017 only 10% were taken from the set of pre-defined configurations.

Finally, the last and the most important reason is the overload acceptance by the FMP due to low-complex traffic situations (over-flights cruising at constant altitude). Sector load analysis of the active sectors in the realized scheme revealed that **although many sectors were overloaded very few flights were actually regulated**. Those tactical decisions are based on the FMP experience, knowledge of the airspace, etc. and are, therefore, hard to model. Since such decision was not part of the ASP optimization algorithm, this created significant difference in the number of active sectors.

Due to all aforementioned reasons, the comparison of the simulated and realized opening scheme was difficult and besides similar tendencies (i.e. increase/decrease in number of active sectors with change of traffic), no other conclusion could be derived.

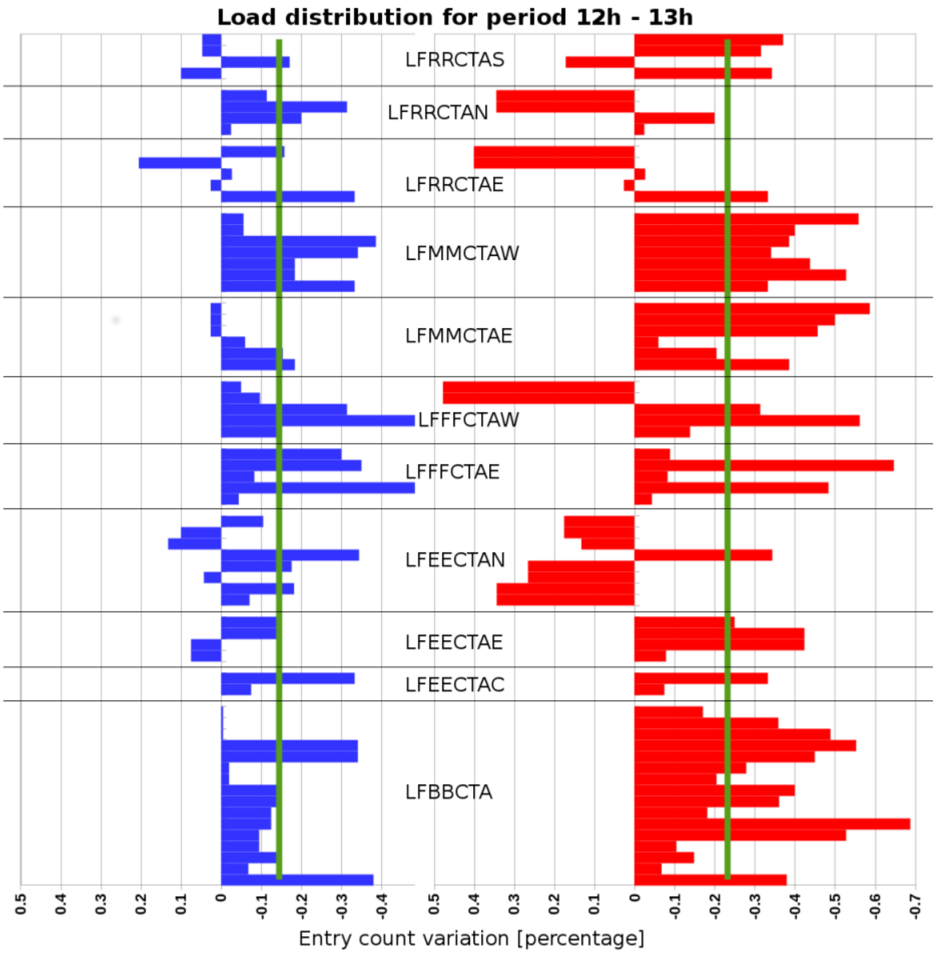


Figure 3-28. Detailed active sector load distribution for one period

As alternative, the next validation exercise considered a comparison of the ASP results with the NEST ICO optimizer. The same date (Feb 20th 2017) and geographical scope (French airspace) were selected, as in the previous exercise. Figure 3-26 shows the distribution of the number of opened positions in the French airspace during the morning peak of the day of study. The ASP results (blue) are compared with those obtained from NEST ICO tool (red) for this validation purpose. As seen in the figure, there is a high matching between ASP and ICO results. However, due to the higher flexibility of the ASP algorithm airspace configuration is better adapted to the traffic resulting in the lower number of the open positions in general.

Figure 3-27 shows the distribution of the active sector load (overload or underload), expressed as variation percentage of entry count from capacity value. For each period, a five-number statistic of the load distribution, including the Interquartile range - IQR (grey bars), for opening schemes provided by the ASP (blue) and ICO (red) are shown. Lower IQR in the opening scheme proposed by ASP signifies smaller dispersion, i.e. more even distribution of the load among active sectors. As shown, load median (black circle Figure 3-27) in the ASP opening scheme is always closer to optimal (zero) value that implies higher capacity utilisation and explains lower number of open positions compared to ICO results.

A more detailed analysis is given in Figure 3-28, showing the distribution of the active sector load for a single period of the same validation exercise. Again, blue bars represent ASP results, while red bars represent ICO results. Green lines in the Figure represent the mean value of the sector load. This Figure confirms once more that, besides having mean value closer to optimal zero value, the ASP opening scheme shows more even distribution of the sector load, represented by smaller deviation of sector loads from the mean value, providing fair distribution of the workload among controllers.

Similar Figures and behaviours are observed for the comparison of the ASP and ICO opening scheme for different days of study, geographical scopes (FR, FABEC) and traffic data sources (historical, synthesised).

Wrapping up, all these exercises confirmed that the optimal opening scheme is computed by the ASP in such way that number of active sectors is minimized, and sector capacity constrains respected. They also confirm that resulting opening scheme reflect current operations and could be used as a valid solution in the operations.

3.2.3 Traffic and capacity planner (TCP) component

The TCP has been validated only when demand and capacity imbalances are solved as in current operational approach: applying delays to avoid sector overloads using the computer assisted slot allocation (CASA) algorithm implemented in the Eurocontrol's CFMU (central flow management unit). The TCP in advanced demand and capacity balance (ADCB) mode has not been validated since the maturity of the algorithm is still very low and only a research prototype is implemented for APACHE.

The file with the description of the airspace is a format agreed between the ASP and the TCP, where the minimum opening scheme time is 20 minutes. This is a limitation imposed during design that has complicated the full validation of the APACHE implementation with that one done in NEST.

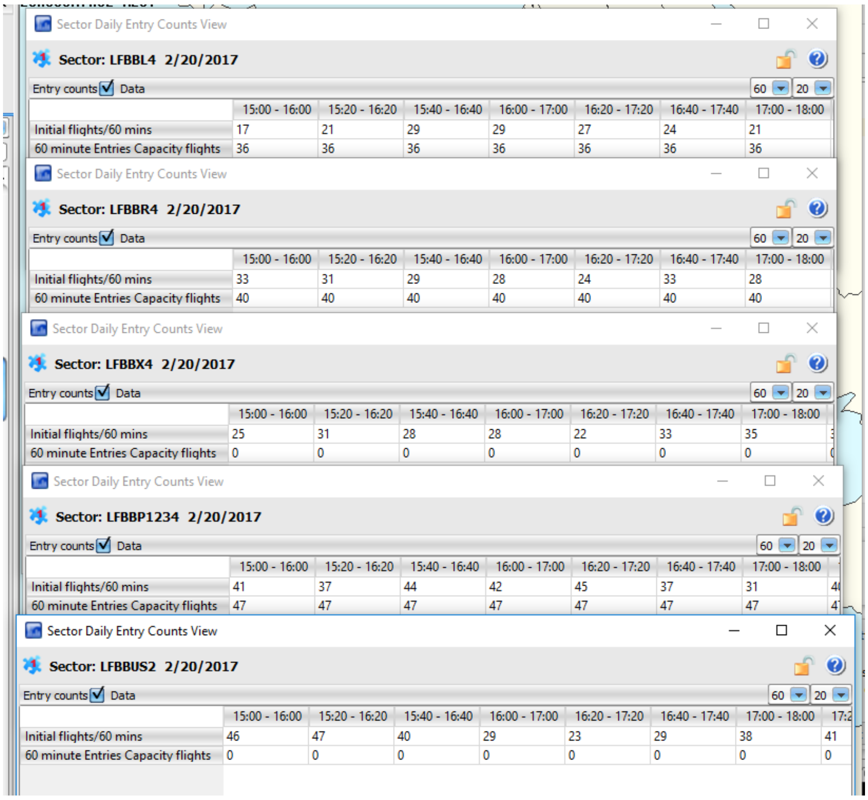
As a first validation, the entry counts per basic sector have been validated with the those numbers given by NEST before any regulation was applied. It was checked that they are exactly the same in numbers and also checked the exact list of concerned flights. The configuration of NEST needs to be the same as in CASA: periods of 1 hour but overlapped every 20 minutes as shown in Figure 3-29.

The second validation aimed to test that the CASA algorithm is indeed keeping the demand below the capacity limits (accounting for eventual sector overloads). 24h of traffic above the French airspace were used in the test and after some rounds of the CASA algorithm all sectors had a demand equal or below the nominal capacity.

The third validation test was done by comparing the results of the APACHE TCP algorithm with the ISA-CASA algorithm available in NEST. For this validation the actual airspace structure of Feb 20th 2017 has been selected, but with the traffic generated by the TP feeding the traffic input of NEST.

The CASA algorithm has been run with 0%, 10% and 20% of posible sector overload (above the sector nominal declared capacity) and the number or regulations, and the number of regulated aircraft have been counted, along with the total number of minutes of delay.

The same numbers are shown for the APACHE TCP CASA algorithm, with the exception that the same airspace structure (comming from the ASP) did not support periods different to multiples of 20 minutes (limitation of the ASP). This will not be a problem for the APACHE simulations foreseen in WP5, since the ASP is used in all the workflow. Yet, the validation results with NEST will show some discrepancies (partially) due to this issue.



Sector: LFBBL4 2/20/2017

	15:00 - 16:00	15:20 - 16:20	15:40 - 16:40	16:00 - 17:00	16:20 - 17:20	16:40 - 17:40	17:00 - 18:00
Initial flights/60 mins	17	21	29	29	27	24	21
60 minute Entries Capacity flights	36	36	36	36	36	36	36

Sector: LFBBR4 2/20/2017

	15:00 - 16:00	15:20 - 16:20	15:40 - 16:40	16:00 - 17:00	16:20 - 17:20	16:40 - 17:40	17:00 - 18:00
Initial flights/60 mins	33	31	29	28	24	33	28
60 minute Entries Capacity flights	40	40	40	40	40	40	40

Sector: LFBBX4 2/20/2017

	15:00 - 16:00	15:20 - 16:20	15:40 - 16:40	16:00 - 17:00	16:20 - 17:20	16:40 - 17:40	17:00 - 18:00
Initial flights/60 mins	25	31	28	28	22	33	35
60 minute Entries Capacity flights	0	0	0	0	0	0	0

Sector: LFBBP1234 2/20/2017

	15:00 - 16:00	15:20 - 16:20	15:40 - 16:40	16:00 - 17:00	16:20 - 17:20	16:40 - 17:40	17:00 - 18:00
Initial flights/60 mins	41	37	44	42	45	37	31
60 minute Entries Capacity flights	47	47	47	47	47	47	47

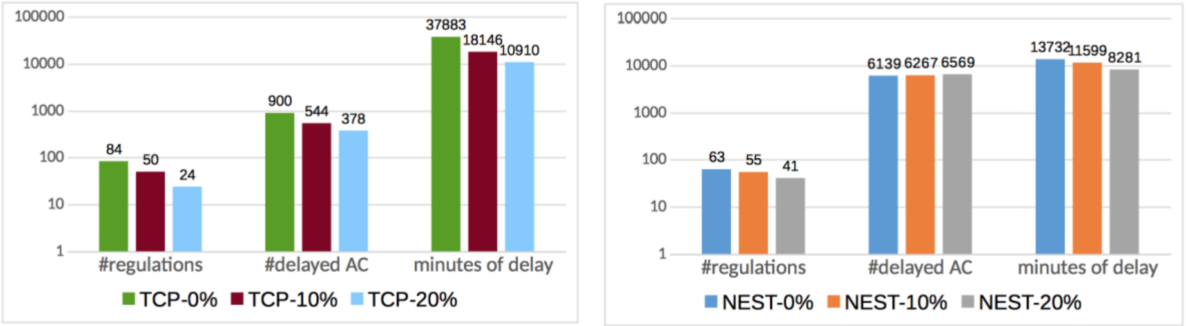
Sector: LFBBUS2 2/20/2017

	15:00 - 16:00	15:20 - 16:20	15:40 - 16:40	16:00 - 17:00	16:20 - 17:20	16:40 - 17:40	17:00 - 18:00	17:2
Initial flights/60 mins	46	47	40	29	23	29	38	41
60 minute Entries Capacity flights	0	0	0	0	0	0	0	0

Figure 3-29. NEST entry counts of 1 hour, given at 20 minute intervals

Figure 3-30 shows the comparison of both results. Notice that the number are not coincident, but the order of magnitude are similar for the umber of regulations, which is around 50 sectors. For this metric (and for the rest too) it can be seen that the TCP is much more sensitive to the overload of demand allowed on top of the capacity.

It is not clear why the NEST ISA-CASA is not so sensible as TCP CASA algorithm. In the number of aircraft observed the NEST algorithm affects much more number of aircraft than what TCP does, but with less delay in total. Part of the differences are given by the fact that the open sectors are not lasting the same (as explained before). Yet, this is not the main reason and no reasonable explanation has been found for these differences, providing that almost not details on the NEST implementations are publicly available.



(a) CASA implemented in APACHE TCP

(b) CASA implemented in NEST ISA

Figure 3-30. Comparison between APACHE CASA implementation and Eurocontrol's NEST

3.2.4 Risk assessment (RA) component

Regarding the **conflict detection module**, a validation exercise was performed using a real-life data: a Comparison with another model - a short validation session was performed (July 28, 2016 sample - 2 hour traffic crossing French airspace), comparing the number of conflicts detected with those obtained by in-house software developed by UPC in the context of another SESAR ER project (RWAKE Consortium, 2017). At the first glance, both results look similar (similar number of conflict identified, but higher in case of RA component).

The TCAS module has been previously validated using real life encounters data (seven encounters from Maastricht airspace in 2009). The aim of validation was to provide evidence on how well the model represents real world ACAS operations, taking into account that the model is developed for the purpose of risk and safety assessment (Netjasov et al, 2013). Among the numerous validation techniques, those accepted for this research, i.e. recognised as best suitable for the available data, were the following:

- Historical data validation: if historical data for the actual system exists, it is used to determine (test) whether the simulation model behaves as the system does.
- Comparison to other models: various outputs of the simulation model being validated are compared to outputs of other simulation models that have been validated.

In order to validate the developed TCAS model, a validation process which requires as input historical data as well as already validated simulation model has been proposed. The real-life data served as a basis for the preparation of input data (see Figure 3-31) for both the TCAS module as well as the InCAS model (EUROCONTROL model which has been well-proven across Europe in TCAS encounters analysis and is already validated). An iterative validation process is proposed based on the abovementioned thinking. At each validation level a modelled case (encounter) is compared with a Control case (which could be from real life (i.e. historical data) or from another model) to determine if the two are sufficiently similar. An application of this validation process to the model of ACAS operations shows that its accuracy of identifying and handling TCAS alerts is similar to that of an existing and already validated TCAS simulation model (Netjasov et al., 2013).

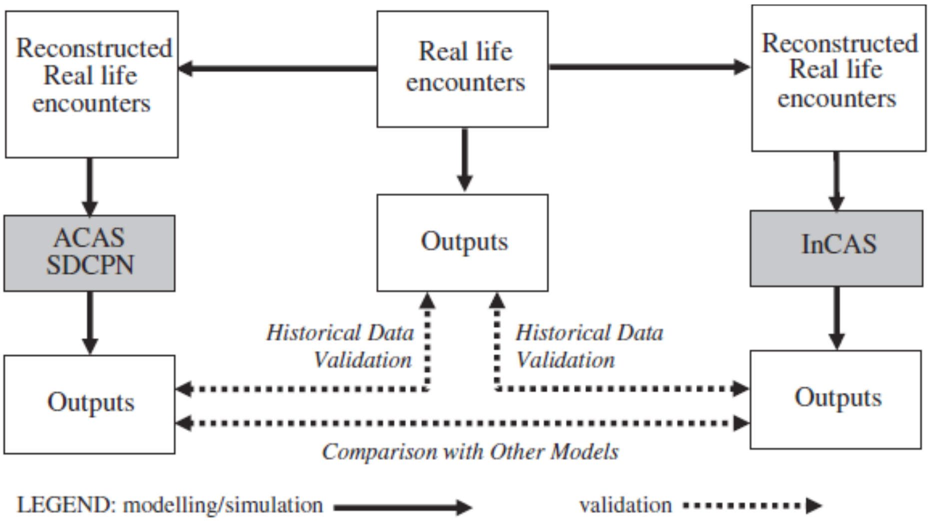


Figure 3-31. TCAS module validation approach (Netjasov et al., 2013)

4 Summary of limitations and assumptions of the APACHE Framework

Taking into account the exploratory nature of the APACHE project and its duration (2 years) the APACHE Framework is built over several assumptions and shows some (known) limitations. Some high-level limitations and assumptions were already outlined in APACHE Deliverable D2.1 (APACHE Consortium, 2017a). For the sake of completeness, these are enumerated again in this document (Section 4.1). Then, during the software requirement and design phases of the APACHE Framework, more specific technical limitations were identified, some of them arising during the implementation and verification stages. These are listed in Section 4.2 and separated per component of the APACHE System.

4.1 General limiations and assumptions

- Only the **en-route** airspace structure is considered (TMA operations and TMA airspace are not modelled).
- Only Instrumental Flight Rules (**IFR**) traffic is considered in the simulations (pre-ops scenarios), neglecting Visual Flight Rules (VFR) traffic.
- All simulated airspace (pre-ops scenarios) is considered for **civil usage** only and therefore segregated airspace or (advanced) flexible use of airspace (A)FUA concepts are not considered.
- Remotely Piloted Aircraft Systems (**RPAS**) and Unmanned Aircraft Systems (UAS) operations are not considered.
- Only **nominal** flight operations are simulated: contingency or emergency procedures are not taken into account.
- Interactions with **airports** are not considered. Thus, all delays due to airport operations are neglected in the simulations (pre-ops scenarios).
- All delays attributable to **airspace users** (such as maintenance issues) are not modelled in the simulations (pre-ops scenarios).

4.2 APACHE-TAP limitations and assumptions

4.2.1 Trajectory planner component

- The lateral route and vertical profile optimisation processes are **decoupled**. That is, DYNAMO first generates the optimal route (sequence of waypoints from origin to destination) by using the wind and fuel flow at “guess” altitude. The fuel flow is computed using a “guess” mass and the optimal cruise speed for these flight conditions. Then, the route is fixed and the optimal vertical profile (altitude and speed) is computed.

- The **lateral route** optimisation is performed only **within the ECAC area**. The route segments outside the ECAC area are fixed to those initially planned by the airspace user.
- A fixed **route structure graph** is used for a given TP simulation. That is, no changes in the Free Route Areas (FRA), conditional airways and direct routes are modelled during a simulation.
- The **FRA** are considered in **2D** and altitude restrictions in the available flight levels are not considered (altitude violations could be detected in post-process).
- The **route structure graph** does not take into account conditional airways, and restrictions in the airways as a function of the cruise altitude, origin/destination airports etc. (as specified in the Route Availability Document).
- The **GRIB** whose validity hour is closest to the departure time of each flight is used to model the weather during the whole trajectory, regardless of the duration of the flight. In other words, the weather is assumed to be a function of latitude, longitude and altitude, but not of time. This assumption would lead to a loss of accuracy for long flights, where the weather at the end of the trajectory will be modelled for that corresponding at the departure time. In addition, convective weather is not considered during the optimisation process.
- For those flights which ICAO code does not match any **aircraft performance model** included in **BADA**, the most similar BADA model has been adopted. This match is performed by using clustering techniques as described in Section 2.1.1.2. This assumption would lead to a loss of accuracy for those trajectories which aircraft model is not directly included in BADA.
- **Helicopters, turbo-propelled aircraft and piston engine** aircraft are not considered either in post-ops or pre-ops scenarios.
- Only those flights with a **requested flight level above FL195** are simulated (in pre-ops) or reproduced (in post-ops).
- For **post-ops scenarios** (reproduced traffic) the **Cost Index** (CI) is estimated from the trajectory as reported in the So6 of the initially filled flight plan (M1 file). The CI is estimated by assuming that the aircraft was flying at the optimal speed in cruise the phase. In practise, the optimal cruise Mach is computed by the on-board Flight Management System (FMS) as a function of the wind conditions, the cruise altitude, the temperature deviation with respect to International Standard Atmosphere Conditions (ISA), the aircraft mass and the CI. The inverse procedure is performed by the APACHE TP to obtain the CI when the optimal Mach and all the other variables are known. The cruise Mach (which is assumed to be optimal) is estimated from the segments at constant altitude of the initially filled flight plan.
- For **pre-ops scenarios** (synthesised traffic), it is assumed that all flights are flying using a **CI** representative of Long Range Cruise (LRC) operations. The CI corresponding to LRC is different for each aircraft model and also depends on the flight conditions (altitude, mass and wind), thus changes along the trajectory. For each aircraft model, the CI corresponding to LRC has been computed off-line for many flight conditions, using analytical optimisation techniques. Then, the resulting experimental distribution of CIs has been fitted with a Gaussian function, quantifying in this way the mean and standard deviation parameters. These parameters for each aircraft model are stored in a dictionary that translates an ICAO code to the corresponding Gaussian function description (see Section 3.1.1). During the process of DYNAMO inputs files generation, the Meta component of the APACHE TP selects a random cost index for each flight, based on the Gaussian distribution corresponding to its aircraft model and using the flight identifier as a seed.
- Lacking from an algorithm capable of accurately **estimating the aircraft mass** from the flight information included in the So6 files, for post-ops scenarios it has been assumed that all the

flights arrive at the destination airport with a fixed landing mass corresponding to 90% of the Maximum Landing Mass (MLM) of the corresponding aircraft model.

- For **pre-ops scenarios,** it has been assumed that all the flights arrive at the destination airport with a landing mass following a Gaussian distribution centred at 90% of the MLM and with a standard deviation of 10%. The flight identifier is used as a seed to generate the random value.

4.2.2 Airspace planner component

- No uncertainty of the demand is considered – it was assumed that planned trajectory data are deterministic, and no trajectory uncertainty was considered during optimization of the sector opening scheme. Furthermore, sector opening scheme was computed once for the whole day and not updated later, since no traffic updated are model neither. Those limitations does not anyhow make ASP result less valuable, but just highlight a difference from the todays operation where opening scheme is constantly updated in respect with new traffic data. This represent another reason why planned (synthesised) and realized opening scheme are hard to compare.
- No configuration of transition preferences – configuration transitions evaluated by the ASP pre-processor where all assigned by equal preferences, meaning equal cost of airspace reconfiguration. As a result, level at which configuration is adapted to given traffic situation solely depend on evaluated sector load and it is not scale by any preference factor. This limitation was imposed, since computation of the configuration transition preferences requires additional operational data usually manually collected based on the FMP and ATCo experience and knowledge of the airspace configurations.
- Bad weather and military activities only impact in a capacity reduction – in the ASP any disruptive event was modelled as a reduction of the capacities of affected sectors.

Specifically, **limitations for the ASP in the DAC mode**:

- It was decided to determine capacity in the terms of complexity by observing historical traffic and airspace opening scheme data. Due to conclusions about reliability of the opening scheme data available in the DDR database during verification and validation tests, it was later decided to take only synthesised data of static scenarios coming from TP and ASP module. Then for the capacity in the DAC mode it was decided to take the highest complexity observed in any active sector that was by the mean of current sector loads (entry counts) considered feasible and respecting capacity limitations. This process considered collection of the eSo6 traffic data and computation of the sector complexities for all available 'static' sectors for all periods of the day. Then considering ASP results of the static scenarios only active sectors for the given activation period are filtered, excluding the ones that had high sector overload (top elementary sectors as explained in 3.1.2). Finally, based on the collected complexity values, a capacity for the DAC algorithm was selected. **This process, not initially considered as important, took considerable amount of time and was a partial reason for the decision to discard DAC scenarios from the WP5 simulations.**
- Current elementary sectors used as SBBs – for the better comparison of the results from static and dynamic scenarios, it was decided to use existing 'static' elementary sector as the sector building block in the DAC. Computation of the capacity in the term of complexity also influenced this choice. However, this choice broth additional difficulties in definition of the sector neighbourhood and generally in the ASP module workflow that now include definition of the sector neighbourhood.

- Finally, the last and the most important limitation for integration ASP in DAC mode into APACHE framework, was **inability to deliver output data into previously established data format**:
    - The first difficulty was due to non-existence of the collapsed sectors in the DAC mode. The liberty to choose any appropriate grouping of the elementary sectors, as main property of the DAC problem classifies DAC into the class of *HP_hard* problems. Although DAC problem was solved without explicitly enumerating possible combinations of the elementary sector, in order to fit results in the main output format *SectorScheme* file, it was required after problem was solved to define and name all grouping of the sector produced by the optimization algorithm.
    - The second and the most important limitation was **capacity information in the *SectorScheme***. The TCP module needs the capacity information in terms of entry counts for each active sector in order to compute hotspots and necessary delays if the actual number of flight entering active sector is greater than sector capacity. However, capacity in the DAC algorithm was given in the terms of complexity and any trial to link those two value and to choose proper value for the integration of the DAC into APACHE framework took too much effort and time, and was finally dismissed due to lack of time.

4.2.3 Traffic and capacity planner component

- The minimum time spent in a sector is 60 seconds.
- The opening scheme is always defined for periods of time that are multiple of 20 minutes.
- The regulations or unballance between demmand and capacity are defined for situations where the entry count is 20% above the nominal declared capacity.
- The entry and exit times of each aircraft in each basic sector is obtained from an external tool (Eurocontrol's NEST). Manual execution is required for it.
- There is no simulation of the submission time of a flight plan into the system. This makes a big difference with current execution of TCP, because the last filled flight plan in the system is the one that most probably will suffer a delay.
- Regulations can be applied to any aircraft crossing the concerned sectors, included those that come from outside of the ECAC area (which are typically excempted of ATFM measures).
- In CASA mode, it is assumed that aircraft with delay shorter than 10 minutes are not regulated because they are within their 15 minutes slot.
- In CASA mode, the slot allocation vector has the length corresponding to the duration of the active sector openning time. When an aircraft gets delayed beyond the end of the slot allocation vector it is always assumed that the slot is free. A new iteration of CASA shall be executed to test new unballances due to these delayed aircraft.
- In ADCB mode, it is assumed that all airspace users are willing to provide alternate trajectories (if operationally possible).
- In ADCB mode, it is assumed that all airspace users are willing to sharespecific information to the Network Manager, such as cost of fuel and time ratios.
- In ADCB mode, the cost of delay is assumed linear.
- In ADCB mode, the unit cost for (different type of) delay is assumed to be the same across different flights.

4.3 APACHE Performance Analyser limitations and assumptions

4.3.1 Risk assessment component

- The RA component is based on the assumption that conflict between pair of aircraft exists when either horizontal and/or vertical separation minima are violated. A separation minima used was: 5 NM horizontal and 1000 ft vertical.
- In cases that conflict situation worsens then the TCAS module is activated. TCAS module contains a TCAS v7.0 model.
- Input trajectories for all flights obtained from Trajectory planner component (*.eSo6 files) have been processed without any change or pre-filtering process.
- Deterministic simulation of flights is performed with time increment of 10 sec.
- In order to speed up search for the conflicts and calculation of safety PIs, a three phase approach is followed: a) Reduction of traffic input (triage) eliminating flights not in conflict (divergent trajectories, different FLs, different entry times, etc.), b) Determination of flights in conflicts and calculation of risks and other safety indicators, and c) Checking whether TCAS will be activated and how.

4.3.2 Computation of PIs

As initially planned, the Performance Analyser component was developed last as part of WP4, which is why some of its limitations and assumptions were identified later in the process of APACHE System development. This required a subsequent modification of certain output files from the main three APACHE TAP components.

Since the majority of the PIs (ENV, CE and CAP ones, in particular) rely on trajectory data obtained from TP and TCP components, an extended So6 (*eSo6*) format (see Appendix C) was agreed in order to provide additional information for the Performance Analyser, such as fuel consumed, route charges, cost index etc. This also implies the necessity for “post-ops” flight data pre-processing in Scenario 0, since the aforementioned additional information is not available in raw *so6* files obtained from DDR2.

All of the PIs listed in the APACHE Deliverable D3.2 (APACHE consortium, 2018), have been finally implemented in the Performance Analyser, **with the exception of AEQ-3, AEQ-4, AEQ-5, CE-1.3 and PAR-1:**

- **AEQ-3 and AEQ-5** are covered by CAP and CE PIs and will be de facto considered later as part of the performance assessment process;
- **AEQ-4** is in fact the intermediary result of AEQ-1 and its values can be found in AEQ-1 *_debug.txt* output file;
- **CE-1.3** implies the difference in route charges for the RBT and SBT trajectories, which is not in accordance with the current CRCO practice of calculating the charges solely based on the last-filed flight trajectory. Moreover, no changes to the current route charging system in Europe are foreseen and its future outlook remains unknown for the time being;
- **PAR-1** was discarded due to data availability reasons**.**

Moreover, some PIs have been implemented with the following limitations:

- **AEQ-1**: This PI is calculated for the whole network and all airlines, even those operating only a few flights, which causes the “max” operand in the formula to be always equal or close to 100% (APACHE, 2018). For detailed analysis per specific airlines it is recommended to use the _*debug.txt* file;
- **CAP-3**: Due to data availability reasons, the PI is currently implemented as the percentage of regulated flights passing through a specific elementary sector and its results are stored solely in the _*debug.txt* file.
- **C-ENV-1 (KEP) and C-ENV-2 (KEA):** The PIs are implemented as the comparison between the length of a trajectory and the shortest (Great Circle) distance between its endpoints (origin/entry point and destination/exit point), but without computing achieved distance and thus without measuring local performance. Also, a 40 NM radius around airports of departure and arrival are not excluded. This limitation is mitigated by the fact that tactical phase of operations is not simulated in APACHE project, which is why no significant impact on these PIs is expected in TMA.
- **FLEX-2:** To estimate capacity utilisation, sector entry counts were used instead of occupancy counts.
- **CE:** Fuel cost and cost of ground delay are assumed to be 0.7€/kg and 49.5€/min respectively, according to (Eurocontrol, 2015) Cost of delay represents the average cost per minute to the airline of tactical ground delay with network effect (including reactionary delay) and this value is recommended to be used for system-wide studies. Although APACHE team is fully aware of the complexity of airline cost modelling, these values are used for the sake of simplification.

5 References

APACHE Consortium. 2017a (Feb). Scope and definition of the concept of operations for the project. APACHE Project, Deliverable D2.1. Version 01.00.00.

APACHE Consortium, 2018 (May). Functional requirements and specifications for the ATM performance assessment framework. APACHE Project, Deliverable D3.2. Version 01.01.00.

Balci, O. 1998. Verification, validation and testing. In: Banks, J. (Ed.), Handbook of Simulation. John Wiley & Sons, pp. 335–393.

Cormen, T., Leiserson,C., Rivest, R. and Stein, C. 2009. Introduction to Algorithms, 3rd Edition. ISBN 978-0-262-03384-8. MIT Press

Dalmau, R., Melgosa, M., Vilardaga, S. and Prats, X. 2018 (Jun). A Fast and Flexible Aircraft Trajectory Predictor and Optimiser for ATM Research Applications. Proceedings of the 8th International Conference on Research on Air Transportation (ICRAT). Castelldefels (Catalonia).

Eurocontrol. 2015 (Nov). Standard Inputs for EUROCONTROL Cost-Benefit Analyses. 7th Ed.

Eurocontrol. 2016 (Jun). DDR2 Reference manual for general users. Version 2.1.3.

Huberty, C. J. 2014 (Sep). Mahalanobis distance. John Wiley & Sons, Ltd.

Hartigan, J. A. and Wong, M. A. 1979. Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society, Series C. Vol. 28 No. 1, pp. 100–108.

Jolliffe, I. 2014. Principal Component Analysis. John Wiley & Sons, Ltd.

MITRE. 2014. Systems Engineering Guide. ISBN 978-0-615-97442-2. The MITRE Corporation.

Netjasov F. 2012. Framework for airspace planning and design based on conflict risk assessment, Part 2: Conflict risk assessment model for airspace tactical planning, *Transportation Research Part C*, Vol. 24, pp. 213–226.

Netjasov F., Vidosavljevic A., Tosic V., Blom H. 2013. Development, Validation and Application of Stochastically and Dynamically Coloured Petri Net Model of ACAS Operations for Safety Assessment Purposes, *Transportation Research Part C*, Vol. 33, pp. 167-195.

Netjasov F. & Crnogorac D. 2018. Potential Safety Occurrences as Indicators of Air Traffic Management Safety Performances: A Network Based Simulation Model. Proceedings of the 8th International Conference on Research in Air Transportation (ICRAT).

Nuic, A., Poles, D., and Mouillet, V. 2010. BADA: An advanced aircraft performance model for present and future ATM systems. Adaptative Control and Signal Processing, vol. 24, pp. 850–866.

Roberson, W., Root, R., and Adams, D. 2008. Fuel Conservation Strategies: Cruise Flight. Boeing Aero Quaterly, The Boeing Company.

RWAKE Consortium. 2017 (Feb). System design document. RWAKE Project, Deliverable D3.1. Version 01.00.00.

Sargent, R. 2009. Verification and validation of simulation models. In: Proceedings of the 2009 Winter Simulation Conference, USA.

Senzig, de. Fleming, G. and Lovinelli, R. 2009. Modelling of Terminal- Area Airplane Fuel Consumption. Journal of Aircraft, vol. 46, no. 4, pp. 1089–1093.

Sergeeva, M., Delahaye, D., Zerrouki, L., and Schede, N. 2015. Dynamic Airspace Configurations Generated by Evolutionary Algorithms. Proceedings of the Digital Avionics Systems Conference (DASC). Prague (Check Republic).

Sergeeva, M., Delahaye, D., Vidosavljevic, A., and Mancel, C. 2017. Dynamic Airspace Configurations by Genetic Algorithm, Journal of Traffic and Transportation Engineering, vol. 4, no. 3, pp. 300-314.

Walker, D. and Dongarra, J. 1996. MPI: a standard message passing interface. Supercomputer, 12. pp. 56–68.

Wandelt. S and Sun. X and Gollnick. V 2014. SO6C: Compressed Trajectories in Air Traffic Management. Air Traffic Control Quarterly, Vol. 22, No. 2, pp. 157-178.

Xu, Y. and Prats, X. 2017 (Oct). Including Linear Holding in Air Traffic Flow Management for Flexible Delay Handling. Journal of air transportation. Vol. 25. No. 4. pp. 123-137.

Xu, Y., Dalmau, R. and Prats, X. 2017. Maximizing airborne delay at no extra fuel cost by means of linear holding. *Transportation Research Part C: Emerging Technologies*, *81*, pp.137-152.

Appendix A Description of the input/output interface filesof the APACHE System

This appendix details and summarises the input/output interface files of the APACHE System. The workflow is presented in Figure 2-1.

A.1 TP input-output files

The inputs and outputs of the APACHE TP for a given combination of Scenario, Area, Case-Study and mode are stored in in the folder <Scenario>/<Area>/<Case-Study>/**TP**/<mode>. This folder contains the following files and/or directories:

- *README:* Text file containing high-level information regarding the status of the TP simulation (number of flights to be optimised, flights filtered using the PCA and Mahalanobis distance criteria, time spent on generating the inputs, time spent on optimising the flights, etc). This file is created by Meta at the end of the simulation.
- *<Case-Study>.log*: Text file with a detailed log of the whole simulation, used for debugging purposes and for tracking eventual warnings and errors. This file is created by Meta.
- *ignoredFlights*: Text file with the list of flights to be ignored from the original demand file (those flights whose requested cruise flight level is below FL195, flights with piston engine aircraft models, or flights to be discarded specifically requested by the user).
- *simulatedFlights: Text file with the list of flights that will be subject to simulation by the TP.*
- *input:* Folder containing one compressed file: TP_debug_<Scenario>_<Area>_<Case-Study>_input.tar.gz: Compressed set of folders (one per flight) with the input files used to run each instance (flight) of the TP. These folders are labelled according to the rank of each flight in the exp2 file.
    - AIRCRAFT.xml: Includes the path to the BADA aircraft performance file and to the optimal cruise, climb and descent speed tables, which are computed off-line.
    - SCENARIO.xml: Includes information about the flight identifier and callsign, the path to the GRIB formatted file containing the weather data, to the file containing information about the route zones and associated charges, and to the binary files encoding the Air Traffic Services (ATS) route network.
    - KNOBS.xml: Includes configuration parameters of the simulation, such as the type of lateral route optimisation (free or structured route), the cost index, the discretisation interval, the numerical integration scheme, etc.
    - CONVENTIONAL.xml: Includes information about the flight phases and associated constraints. This file also includes the initial and terminal conditions (landing mass, origin and destination airport coordinates, departure time, etc.)
- *output*: Folder containing three compressed files:
    - *TP_debug_<Scenario>_<Area>_<Case-Study>_original.tar.gz:* Compressed set of folders (one per flight) for debugging purposes and with some detailed trajectory results. These folders are labelled according to the rank of each flight in the exp2 file.
    - *TP_raw_<Scenario>_<Area>_<Case-Study>_original.eSo6.tar.gz:* Compressed concatenated eSo6 file with all simulated flights (still subject to outliers and inconsistencies).
    - *TP_output_<Scenario>_<Area>_<Case-Study>_original.eSo6.tar.gz:* Compressed concatenated eSo6 file with all these outliers removed. **This is the TP System interface file that will serve as input by the ASP, TCP and PA** (see Figure 2-1).

A.2 ASP input-output files

The inputs and outputs of the APACHE TCP for a given combination of Scenario, Area, Case-Study and mode are stored in in the folder <Scenario>/<Area>/<Case-Study>/**ASP**/<mode>. This folder contains the following files and/or directories:

- ASP_output_<Scenario>_<Area>_<Case-Study>_<mode>.log: Text file containing high-level information regarding the status of the airspace optimization: scenario, area and case study information, location of input/output files, number of input flights presented in the input t5 file, optimization period and other optimization parameters, statuses and warnings of the optimization process phases, etc.
- input: Folder with the main input files. Intermediate files presented in section 2.1.2 that are not crucial for the APACHE workflow, were omitted:
    - Traffic input files:
        - TP_output_<Scenario>_<Area>_<Case-Study>_original.eSo6.tar.gz: symbolic link pointing to TP standard output file used as main input file for the ASP.
        - TP_output_<Scenario>_<Area>_<Case-Study>_original.t5: NEST Airspace-Traffic intersection file, built based on the eSo6, used by ASP to calculate entry counts of the sectors.
    - Airspace input files:
        - ES: File in the ASP format containing list of the elementary sectors for the concerned area of the specific case study.
        - CS: File in the ASP format containing list of the collapsed sectors for the concerned area of the specific case study. Each sector contains definition i.e. list of elementary sectors it contains.
        - CONFIG: File in the ASP format containing list of the airspace configurations for the concerned area of the specific case study. Each configuration contains name of the cluster (ACC) it belongs and list of sectors (elementary or collapsed) it is built of.
        - TRANSITION_CONF: File in the ASP format containing list of feasible transitions between airspace configurations
        - CLUSTER: File in the ASP format containing list of the clusters and/or ACC in the concerned area of the specific case study.
        - ENTRY_COUNTS: File in the ASP format containing list of control sectors (elementary or collapsed) with assigned capacity in the terms of entry counts and period of its validity. Validity period could be used to simulate special event like weather, strikes, etc. when naturally sector capacity could be reduced for certain periods.
- *output*: Folder containing output files:
    - ASP_output_<Scenario>_<Area>_<Case-Study>_<mode>.SectorScheme: lists all elementary sectors with active sectors for each period in which they are belonging. The main output file, representing ASP System interface that serves as input to other modules, such as the TCP and PA (see Figure 2-1). Detailed format description of SectorScheme file is given in the Appendix D.
    - ASP_output_<Scenario>_<Area>_<Case-Study>_<mode>.sRE: Sector opening scheme output file containing, for each period, the number and the list of active sectors.

  - ASP_output_<Scenario>_<Area>_<Case-Study>_<mode>.cos: Sector opening scheme output file in the NEST format containing, for each period, list of active airspace configurations grouped by clusters/ACCs.
  - ASP_output_<Scenario>_<Area>_<Case-Study>_<mode>: Detailed sector opening scheme output file. It contains for each period the total number of active sectors and for each cluster/ACC a number and list of active sectors attached with sector load.
  - ASP_output_<Scenario>_<Area>_<Case-Study>_<mode>.ENTRYRATE: lists all active sectors with activation time, number of aircrafts entering in the sector in next hour and capacity of the sector in the terms of entry counts for that hour. It is manly used for debugging.

A.3 TCP input-output files

The inputs and outputs of the APACHE TCP for a given combination of Scenario, Area, Case-Study and mode are stored in in the folder <Scenario>/<Area>/<Case-Study>/**TCP**/<mode>. This folder contains the following files and/or directories:

- *input:* Folder containing:
  - A link to the *TP_output_<Scenario>_<Area>_<Case-Study>_<mode>.eSo6* output file of the TP, which contains the whole set of trajectories (demand).
  - A link to the *ASP_output_<Scenario>_<Area>_<Case-Study>_<mode>.SectorScheme* output file of the ASP, which contains sector opening schemes and capacity information.
  - A link to the *TP_output_<Scenario>_<Area>_<Case-Study>_<mode>.t5* output file of the ASP, containing airspace/trajectory intersection information.
  - TCP_input_<Scenario>_<Area>_<Case-Study>_sectors_<AIRAC-date>.are: DDR2 standard file containing sector geometric and geographical information.
  - TCP_input_<Scenario>_<Area>_<Case-Study>_sectors_<AIRAC-date>.sls: DDR2 standard file containing traffic volume information.
  - A link to the TP_output_<Scenario>_<Area>_<Case-Study>_TCP-LAT.eSo6.tar.gz output file of the TP, which contains the lateral trajectory alternatives to solve the ADCB problem (ATFM re-routings). *Only for those scenarios with the TCP in ADCB mode.*
  - A link to the TP_output_<Scenario>_<Area>_<Case-Study>_TCP-VERT.eSo6.tar.gz output file of the TP, which contains the vertical trajectory alternatives to solve the ADCB problem (level cappings). *Only for those scenarios with the TCP in ADCB mode.*
- *output:* Folder containing:
  - *README*: Text file containing high-level information regarding the status of the TCP simulation and a high-level view of the results (total number of regulated aircraft, total delay, execution times, etc.)
  - *TCP_output_<Scenario>_<Area>_<Case-Study>_<mode>.Entryrate.csv*: CSV file containing the pre-regulation entryrate and the corresponding capacity for each operating sector in each concerned time period.
  - *TCP_output_<Scenario>_<Area>_<Case-Study>_<mode>.post_entryrate.csv*: CSV file containing the post-regulation entryrate and the corresponding capacity for each operating sector in each concerned time period.
  - *TCP_output_<Scenario>_<Area>_<Case-Study>_<mode>.REGULATIONS:* Text file (**only in CASA mode**) with the information of delays per sector, instead of delays per aircraft. Only the active sectors with higher demand than capacity are listed. The first three columns are given to identify an active sector. This requires: the name of the collapsed sector, the time

period start and the number of periods (20') of the regulation. Then the file provides the numbers of capacity and demand of such slot and the number of delayed aircraft and the total minutes of delay.

- *TCP_output_<Scenario>_<Area>_<Case-Study>_<mode>.HotSpots.tar.gz:* compressed set of files (one per flight crossing a hotspot, **only in ADCB mode**), which contains information on the concerned flight and the geometric characteristics of the sector to be avoided. **This is the main interface file between the TCP and TP in ADCB mode.** The TP will compute alternative trajectories (re-routings and level capping) for each concerned flight avoiding the different hotspots.
- *TCP_output_<Scenario>_<Area>_<Case-Study>_<mode>.Options.csv*: CSV file (**only in ADCB mode**) containing the trajectory option(s), for each flight, that were considered in the ADCB algorithm (i.e., re-routing and/or level capping).
- *TCP_output_<Scenario>_<Area>_<Case-Study>_<mode>.ACdelays.csv:* CSV file containing the specific regulation(s) imposed on each flight, such as the total amount of delay (both in CASA and ADCB modes) and the type of trajectory option (original, re-routing or level capping) that the flight is assigned with (only in ADCB mode). In ADCB mode this file also specifies how the delay is shared among ground delay, linear holding and air delay.
- *TCP_output_<Scenario>_<Area>_<Case-Study>_original.eSo6.tar.gz*: Compressed concatenated eSo6 file with the regulated (and non-regulated) trajectories. **This is the TCP System interface file that will serve as input by the PA** (see Figure 2-1).

A.4 Performance Analyser input-output files

The **outputs** of the APACHE Performance Analyser for a given combination of Scenario, Area, Case-Study and mode are stored in in the folders <Scenario>/<Area>/<Case-Study>/**RA**/ and <Scenario>/<Area>/<Case-Study>/**PA**/. The former contains specific outputs of the risk assessment (RA) module (SAF indicators), while the latter contains all other performance indicators.

The <Scenario>/<Area>/<Case-Study>/**RA**/ folder contains the following files:

- *RA_<Scenario>_<Area>_<Case-study>_Debugging.txt:* Text file with several lines of results. Each line contains individual conflicting flight identifiers, time of CPA (closes point of approach), locations of aircraft in CPA, flight levels at CPA, shortest distance between aircraft in conflict, duration of conflict, severity of conflict and calculated risk of conflict.
- *SAF-<X>_<Scenario>_<Area>_<Case-study>_output.txt:* Text file containing absolute and normalized values for all Safety PIs (<X> from 1 to 7).

The <Scenario>/<Area>/<Case-Study>/**PA**/ folder contains the following files:

- *<PI-code>_<Scenario>_<Area>_<Case-study>_output.txt:* Text file with one or several lines of results. Each line has two columns: a “string” and a “number”. The string is a descriptive text of what the “number” represents. For instance, “average", “median”, “IQR”, “Q1”, … or just “PI” indicating that the value corresponds to the computation of the PI as defined in D3.2 (APACHE, 2018).
- *<PI-code>_<Scenario>_<Area>_<Case-study>_debug.txt:* Text file with relevant data for debugging or detailed analysis purposes. It might contain metrics per flight (not aggregated), intermediate results used to compute the Performance Indicator, etc. The format is ad-hoc for each PI and it is specified in the header of the text file.

As explained in Section 2.1.4 (see also Figure 2-1), all output results of the PA are merged into a single CSV file to facilitate the analysis and benchmarking (to be done in WP5). Table A-1 shows the format and an example of contents of this unified CSV file, which is the main output of the whole APACHE Framework workflow.

Scenario	Case Study	KPA	PI	Metric	Value
S1	S101	ENV	ENV-1.1	Mean	90.2
S1	S101	ENV	ENV-1.1	Median	77.4
S1	S101	ENV	ENV-1.1	IQR	40.1
…	…	…	…	…	…
S1	S101	ENV	ENV-1.2	Mean	20.0
S1	S101	ENV	ENV-1.2	Median	13.2
…	…	…	…	…	…
S1	S101	CAP	CAP-1	PI	55
S1	S101	CAP	CAP-1	Std	6
S1	S101	CAP	CAP-2	PI	78
…	…	…	…	…	…
S1	S101	SAF	SAF-1	PI	327
S1	S101	SAF	SAF-1	PI-norm	0.015
S1	S101	SAF	SAF-2	PI	666
S1	S101	SAF	SAF-2	PI-norm	0.077
…	…	…	…	…	…
S1	S101.PF1a	ENV	ENV-1.1	Mean	131.2
…	…	…	…	…	…
S2	S201	ENV	ENV-1.1	Mean	60.2
…	…	…	…	…	…

Table A-1 Format and example of the final CSV file (output of the whole APACHE Framework workflow).

Appendix B High Performance Computing for the APACHE Trajectory Planner

In order to speed up the computation of multiple aircraft trajectories, within the APACHE module TP (Trajectory Planner), a High-Performance Computing (HPC) cluster has been deployed. This cluster consists of one master/slave node plus several slave nodes connected via local area network (LAN) and installed with a set of software applications such as Open LDAP, NFS, etc. Such software applications have the objective of creating a scalable multi-user/multi-purpose infrastructure. The master node is in charge of coordinating the slave nodes, monitoring their status (e.g. RAM, processors or computing cores utilization, etc.) and it generally implements a set of policies to optimise the resources utilization.

In order, to exploit a HPC cluster, software running on it must be written in a parallel approach, for example using standard parallel libraries such as Open MPI, an implementation of the Message Passing Interface standard (Walker and Dongarra, 1996). In this context, a python software called **Meta** has been developed, which given a set of *N* computing cores, launches the calculation of *N* aircraft trajectories in parallel given a number of available slave nodes. In turn, each slave node includes a **core splitter**, which, once a flight chunk is received, checks the available computing cores in the node and launch a parallel trajectory prediction/optimisation execution assigning one flight to one core (i.e. being *F* the quantity of cores of the node, *F* flights will be processed in parallel). Successive rounds are performed until all the flights in the chunk are processed.

Thus, Meta is scalable to process any quantity of trajectories on any quantity of computing cores and has been possible in the context of the APACHE Project to provide time efficiency and scalability for generating traffic scenarios with a large number of trajectories. Figure B-1 illustrates this HPC architecture.

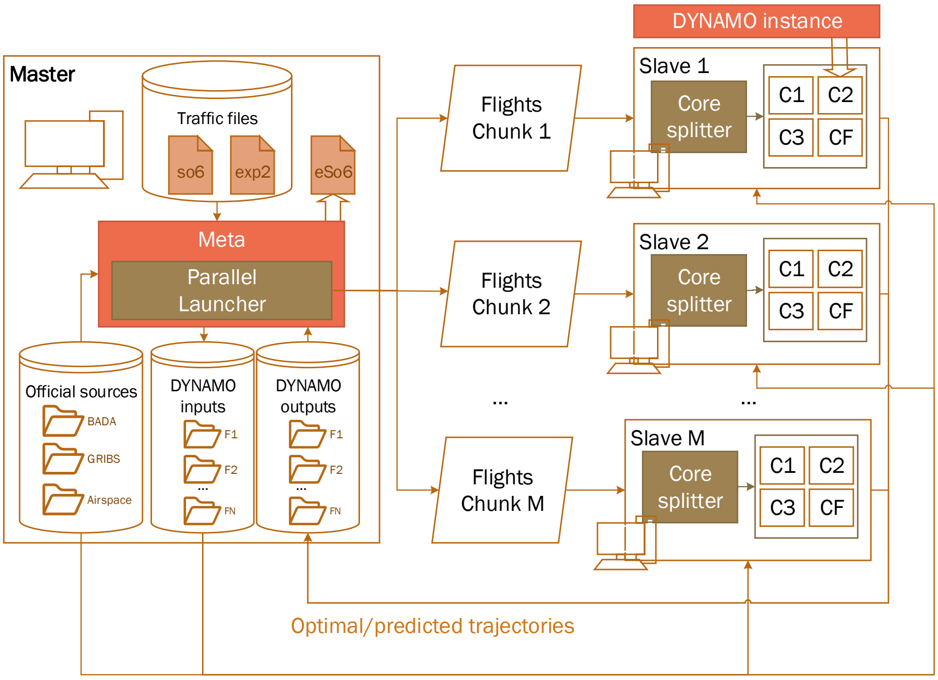


Figure B-1 Schematic view of the APACHE TP high performance computing (HPC) architecture capable to optimise/recreate several trajectories in parallel

Appendix C Extended So6 format

The *eSo6* is an ad-hoc file specification for the purposes of the APACHE project that has the same format and information as a standard *So6* file, as provived in Eurocontrol's DDR2 database (Eurocontrol, 2016), but with some extra information included in 7 extra columns at the end of each line (flight segment).

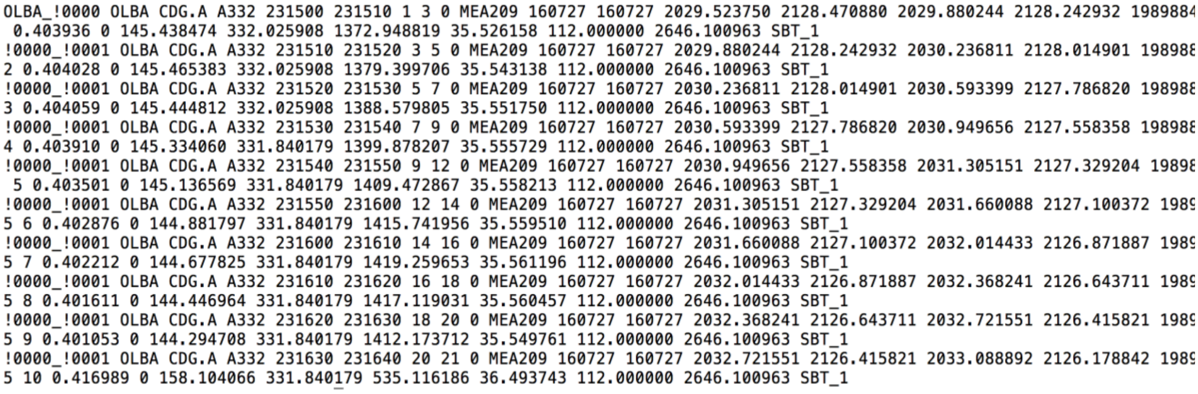
It should be noted that transforming an *eSo6* file to a standard *So6* file (readable by NEST, for instance) is straightforward, since it is only required to cut the last 7 columns of the *eSo6*. This facilitates the usage of NEST (or other third party programs accepting *So6* as input files for trajectory data) for validation or visualisation purposes, and enables the possibility to easily exchange trajectory data from APACHE with other institutions or SESAR Projects.

Table C-1 details the format of the eSo6 file, while Figure C-1 shows an extract of an *eSo6* example file.

File Column	Units	Description	Remarks
1		Segment identifier	<Name of first waypoint of segment>_<name of last waypoint of segment>
2		Origin of flight	ICAO code of origin airport; e.g. EDDF for Frankfurt. All flight segments of a flight show the same value.
3		Destination of flight	ICAO code of destination airport; e.g. EDDF for Frankfurt. All flight segments of a flight show the same value.
4		Aircraft type	ICAO code of aircraft type; e.g. A388 for Airbus A380. All flight segments of a flight show the same value.
5		Time begin	Time of entering the segment; format HHMMSS, padded with 0 from the left.
6		Time end	Time of leaving the segment; format HHMMSS, padded with 0 from the left.
7	FL	FL begin	Flight level (in hundreds of feet) when entering the segment.
8	FL	FL end	Flight level (in hundreds of feet) when leaving the segment.
9		Status	0=climb, 1=cruise, 2=descent
10		Callsign	Call sign with ICAO code for airline; DLH6PH for a Lufthansa flight. All flight segments of a flight show the same value.
11		Date begin	Date of entering the segment; format yymmdd, padded with 0 from the left
12		Date end	Date of leaving the segment; format yymmdd, padded with 0 from the left
13	min	Latitude begin	Latitude in minute decimal of the segment begin; e.g. 3002 for 50°1’60’’ N
14	min	Longitude begin	Longitude in minute decimal of the segment begin; e.g. 514.233 for 8°34’14’’ E
15	min	Latitude end	Latitude in minute decimal of the segment end
16	min	Longitude end	Longitude in minute decimal of the segment end
17		Flight identifier	Unique identifier for the flight; e.g. 172874110. All flight segments of a flight show the same value

File Column	Units	Description	Remarks
18		Sequence	Incremental at each flight segment; e.g. 3 for the third segment of a flight
19	NM	Segment length	Length of the flight segment in nautical miles
20		Segment parity/colour	SAAM-specific colour encoding (values 0-9)
21*	kt	Ground speed	Ground speed for this particular flight segment.
22*	deg	Track	Track speed for this particular flight segment.
23*	ft/min	Rate of Climb	Rate of climb for this particular flight segment. A negative value means the aircraft is descending.
24*	kg	Fuel	Fuel consumed in this particular flight segment.
25*	kg/min	Cost Index (CI)	Cost Index for this particular flight segment (typically the same for all segments, since a CI is assigned for a particular flight).
26*	Eur	Route Charges Cost	Cost for the whole flight (not this particular flight segment). All flight segments of a flight show the same value.
27*	-	Trajectory type	String that can contain the following values: • **SBT_1:** to indicate the trajectory is the first SBT (i.e. the original output from the TP). • **SBT_n:** to indicate the n-th SBT trajectory in the trajectory negotiation process with the network manager, according to the TBO paradigm. In APACHE, this negotiation is not simulated. Instead, the TP is always providing to the TCP with two extra options to avoid hotspots (if that particular flight is concerned): avoiding congested sectors laterally (SBT_1) or avoiding them vertically (SBT_2). This field, however, leaves the door open to future implementations of the TCP with a more complex SBT negotiation. • **RBT:** to indicate an agreed trajectory after the negotiation (i.e. the output of the TCP), in the case that NO regulation is applied (i.e. RBT=SBT_1). • **RBT***: to indicate an agreed trajectory after the negotiation (i.e. the output of the TCP), in the case that a regulation is indeed applied (i.e. RBT != SBT_1). This regulation could be simply delay (after a demand and capacity balance using CASA) or might involve re-routings or level capping. • **RBT_n:** to indicate the n-th RBT trajectory in the trajectory negotiation process with the ANSP, according to the TBO paradigm, once the flight is airborne (tactical trajectory updates). In APACHE, the tactical layer is not modelled but this specification leaves the door open to future implementations of the System. • **ERBT:** Executed RBT (trajectory actually flown)

Table C-1 Extended So6 (*eSo6*) file format



```
OLBA_!0000 OLBA CDG.A A332 231500 231510 1 3 0 MEA209 160727 160727 2029.523750 2128.470880 2029.880244 2128.242932 1989884
 0.403936 0 145.438474 332.025908 1372.948819 35.526158 112.000000 2646.100963 SBT_1
!0000_!0001 OLBA CDG.A A332 231510 231520 3 5 0 MEA209 160727 160727 2029.880244 2128.242932 2030.236811 2128.014901 198988
2 0.404028 0 145.465383 332.025908 1379.399706 35.543138 112.000000 2646.100963 SBT_1
!0000_!0001 OLBA CDG.A A332 231520 231530 5 7 0 MEA209 160727 160727 2030.236811 2128.014901 2030.593399 2127.786820 198988
3 0.404059 0 145.444812 332.025908 1388.579805 35.551750 112.000000 2646.100963 SBT_1
!0000_!0001 OLBA CDG.A A332 231530 231540 7 9 0 MEA209 160727 160727 2030.593399 2127.786820 2030.949656 2127.558358 198988
4 0.403910 0 145.334060 331.840179 1399.878207 35.555729 112.000000 2646.100963 SBT_1
!0000_!0001 OLBA CDG.A A332 231540 231550 9 12 0 MEA209 160727 160727 2030.949656 2127.558358 2031.305151 2127.329204 19898
 5 0.403501 0 145.136569 331.840179 1409.472867 35.558213 112.000000 2646.100963 SBT_1
!0000_!0001 OLBA CDG.A A332 231550 231600 12 14 0 MEA209 160727 160727 2031.305151 2127.329204 2031.660088 2127.100372 1989
5 6 0.402876 0 144.881797 331.840179 1415.741956 35.559510 112.000000 2646.100963 SBT_1
!0000_!0001 OLBA CDG.A A332 231600 231610 14 16 0 MEA209 160727 160727 2031.660088 2127.100372 2032.014433 2126.871887 1989
5 7 0.402212 0 144.677825 331.840179 1419.259653 35.561196 112.000000 2646.100963 SBT_1
!0000_!0001 OLBA CDG.A A332 231610 231620 16 18 0 MEA209 160727 160727 2032.014433 2126.871887 2032.368241 2126.643711 1989
5 8 0.401611 0 144.446964 331.840179 1417.119031 35.560457 112.000000 2646.100963 SBT_1
!0000_!0001 OLBA CDG.A A332 231620 231630 18 20 0 MEA209 160727 160727 2032.368241 2126.643711 2032.721551 2126.415821 1989
5 9 0.401053 0 144.294708 331.840179 1412.173712 35.549761 112.000000 2646.100963 SBT_1
!0000_!0001 OLBA CDG.A A332 231630 231640 20 21 0 MEA209 160727 160727 2032.721551 2126.415821 2033.088892 2126.178842 1989
5 10 0.416989 0 158.104066 331.840179 535.116186 36.493743 112.000000 2646.100963 SBT_1
```

Figure C-1 Extract of an *eSo6* example file

Appendix D SectorScheme format

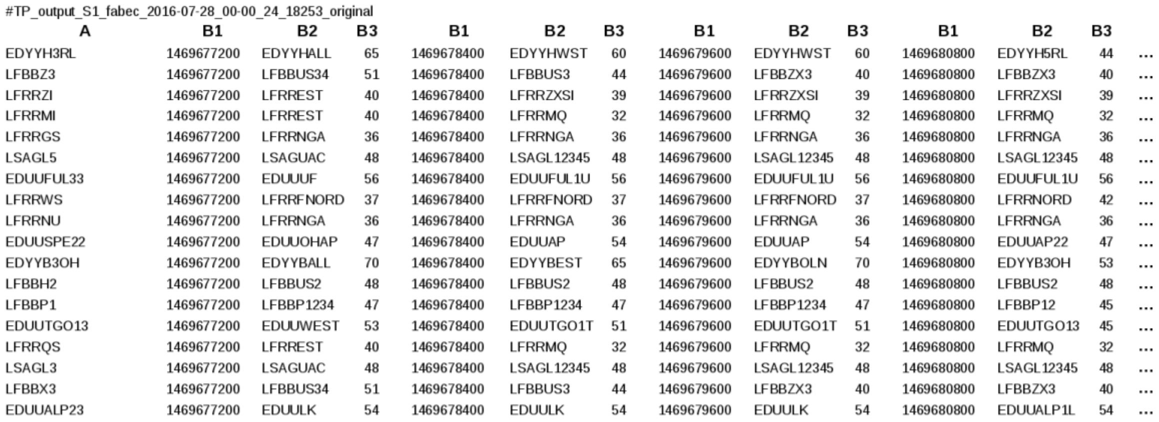
The *SectorScheme* is a file specification specially developed for the purposes of the APACHE project that groups information about airspace sector opening scheme i.e. list of active sectors during simulation period and sector capacities needed for the TCP module and computation of the flight restrictions (delays, reroutings, etc.).

It lists all elementary sectors presented in the studied area (FABEC) and for each period gives active sector in which that elementary sector belongs followed by its capacity. This facilitate data pre-processing of TCP module that would additionally require airspace structure data (ES, CS, CONFIG, ENTRYRATE, files) to extract information needed for the algorithm, if standard cos file was used as interface instead. This way all information needed by TCP module (in addition to t5 traffic file) are provided in a single source.

Table D-1 details the format of the *SectorScheme* file. First comment line contains case study for which file is produced. Three B columns are repeated as many time as there are periods in which the sectorization changes (usually every 20 minutes during the period of the case study of 24h). Figure D-1 shows an extract of a *SectorScheme* example file.

File Column		Units	Description	Remarks
A		char	Elementary sector ID	Unique standardized elementary sector name/ID
B	**1**	int	Activation time	In seconds from the epoch
B	**2**	char	Active sector ID	Standard name/ID of active sector in which elementary sector (column A) belongs at time column B1.
B	**3**	int	Active sector capacity	Active sector (column B2) capacity in terms of entry count at time column B1.

Table D-1 *SectorScheme* file format



#TP_output_S1_fabec_2016-07-28_00-00_24_18253_original

A	B1	B2	B3	B1	B2	B3	B1	B2	B3	B1	B2	B3	
EDYYH3RL	1469677200	EDYYHALL	65	1469678400	EDYYHWST	60	1469679600	EDYYHWST	60	1469680800	EDYYH5RL	44	...
LFBBZ3	1469677200	LFBBUS34	51	1469678400	LFBBUS3	44	1469679600	LFBBZX3	40	1469680800	LFBBZX3	40	...
LFRRZI	1469677200	LFRREST	40	1469678400	LFRRZXSI	39	1469679600	LFRRZXSI	39	1469680800	LFRRZXSI	39	...
LFRRMI	1469677200	LFRREST	40	1469678400	LFRRMQ	32	1469679600	LFRRMQ	32	1469680800	LFRRMQ	32	...
LFRRGS	1469677200	LFRRNGA	36	1469678400	LFRRNGA	36	1469679600	LFRRNGA	36	1469680800	LFRRNGA	36	...
LSAGL5	1469677200	LSAGUAC	48	1469678400	LSAGL12345	48	1469679600	LSAGL12345	48	1469680800	LSAGL12345	48	...
EDUUFUL33	1469677200	EDUUUF	56	1469678400	EDUUFUL1U	56	1469679600	EDUUFUL1U	56	1469680800	EDUUFUL1U	56	...
LFRRWS	1469677200	LFRRFNORD	37	1469678400	LFRRFNORD	37	1469679600	LFRRFNORD	37	1469680800	LFRRNORD	42	...
LFRRNU	1469677200	LFRRNGA	36	1469678400	LFRRNGA	36	1469679600	LFRRNGA	36	1469680800	LFRRNGA	36	...
EDUUSPE22	1469677200	EDUUOHAP	47	1469678400	EDUUAP	54	1469679600	EDUUAP	54	1469680800	EDUUAP22	47	...
EDYYB3OH	1469677200	EDYYBALL	70	1469678400	EDYYBEST	65	1469679600	EDYYBOLN	70	1469680800	EDYYB3OH	53	...
LFBBH2	1469677200	LFBBUS2	48	1469678400	LFBBUS2	48	1469679600	LFBBUS2	48	1469680800	LFBBUS2	48	...
LFBBP1	1469677200	LFBBP1234	47	1469678400	LFBBP1234	47	1469679600	LFBBP1234	47	1469680800	LFBBP12	45	...
EDUUTGO13	1469677200	EDUUWEST	53	1469678400	EDUUTGO1T	51	1469679600	EDUUTGO1T	51	1469680800	EDUUTGO13	45	...
LFRRQS	1469677200	LFRREST	40	1469678400	LFRRMQ	32	1469679600	LFRRMQ	32	1469680800	LFRRMQ	32	...
LSAGL3	1469677200	LSAGUAC	48	1469678400	LSAGL12345	48	1469679600	LSAGL12345	48	1469680800	LSAGL12345	48	...
LFBBX3	1469677200	LFBBUS34	51	1469678400	LFBBUS3	44	1469679600	LFBBZX3	40	1469680800	LFBBZX3	40	...
EDUUALP23	1469677200	EDUULK	54	1469678400	EDUULK	54	1469679600	EDUULK	54	1469680800	EDUUALP1L	54	...

Figure D-1 Extract of a *SectorScheme* example file

Appendix E Fulfilment of the APACHE System requirements

APACHE deliverable D3.2 (APACHE consortium, 2018) presented all requirements of the APACHE system, as first step towards the implementation of the APACHE Framework. This appendix enumerates again all the requirements identified in D3.2 and reports on their fulfilment giving evidences and/or remarks when appropriate.

E.1 Functional requirements

Software Requirement ID	Description	Fulfilment	Evidences/remarks
TP-FR-001	The TP will receive a set of *P2P flight demand* per simulation scenario (flights set), weather data and trajectory constraints (if any) as input to which optimal trajectories will be calculated.	DONE	See section 2.1.1 and Appendix A for a description of the TP input-output files.
TP-FR-002	The information of each *P2P flight* will include origin airport and destination airport described by their geographic coordinates (longitude, latitude) and elevation.	DONE	Included in the EXP2 and So6 files used as traffic demand input of the TP. See section 2.1.1 and Appendix A.
TP-FR-003	The information of each *P2P flight* will include *estimated time of departure* (ETD) and if flight time is fixed and not subject to optimisation (trying to reproduce historical data, for instance) it will also include the *estimated time of arrival* (ETA).	DONE	
TP-FR-004	The information of each *P2P flight* will include the aircraft type and callsign.	DONE	
TP-FR-005	The TP will use specific aircraft performance models according to the aircraft type.	DONE	Using BADA 4.2 data. Licence granted to UPC. See Section 2.1.1.2.
TP-FR-006	The TP will use specific *cost indexes* (CI) according to the airline to which each flight belongs, and eventually also depending on the O/D pair. Since this information is not publicly available, these cost indexes will be assumed/estimated according to assumptions regarding the type of airline (e.g., low-cost) and the ETA if available.	DONE	CIs have been derived empirically for each aircraft model according to (Roberson et al., 2008). See section 3.1.1.
TP-FR-007	The TP will compute a *4D trajectory* for each flight.	DONE	See section 3.1.1 and 3.2.1 for example results arising from the verification and validation tests.
TP-FR-008	The TP will simulate a set of flights under specific ConOps: structured route or free route and flight level allocation/orientation schemes or continuous cruise climb procedures.	DONE	See section 3.1.1 and 3.2.1 for example results arising from the verification and validation tests. See figure 3-2.
TP-FR-009	The TP will use weather information to be considered in the calculation of optimal trajectories per flight.	DONE	See 2.1.1.2 and example results of sections 3.1.1 and 3.2.1.
TP-FR-010	The TP will use specific *Payload weights* according to the airline to which each flight belongs, and eventually also depending on the O/D pair. Since this information is not publicly available, these weights will be assumed/estimated according to assumptions regarding the type of airline (e.g. low-cost) and historical trajectory data (if available).	DONE	Payloads have been deduced from educated guesses. See section 3.1.1.

TP-FR-011	The TP will also consider other trajectory constraints in the form of controlled times of arrival/departure, speed/altitude constraints, control time of arrivals (CTAs), etc.	DONE	See example results in sections 3.1.1 and 3.2.1. See Figure 3-2 and 3-3.
TP-FR-012	The TP shall be able to produce alternative trajectories to avoid one or several hotspots. Alternative trajectories should account for i) lateral avoidance and ii) vertical avoidance of the concerned sector(s).	DONE	See section 3.1.3.2 and in particular Figure 3-16 and Table 3-3.
TP-FR-013	The TP shall be able to compute the best vertical profile (altitude and speed profiles) given an input route.	DONE	See example results in sections 3.1.1 and 3.2.1.
TCP-FR-001	The TCP will receive a set of 4D trajectories per simulation scenario (trajectories set computed by the TP) and an opening scheme (list of active sectors as a function of the time, provided by the ASP).	DONE	See section 2.1.3 and Appendix A for a description of the TCP input-output files.
TCP-FR-002	The TCP will implement an ATFM slot allocation mechanism based on the CFMU CASA algorithm. The demand will be given by the TP, while the sectorisation and nominal capacities per sector will be given by the ASP module.	DONE	See section 3.1.3.1 for the verification and system integration test and 3.2.3 for the validation test.
TCP-FR-003	The TCP will detect hotspots (sectors with demand above capacity) and those flights crossing these hotspots.	DONE	See section 3.1.3.2.
TCP-FR-004	The TCP will implement an advanced demand and capacity balance (ADCB) algorithm which will take into account not only delay as possible measure to shift demand, but also lateral and vertical re-routings (i.e. alternative trajectories avoiding the list of hotspots). This ADCB algorithm will compute a system-wide optimal solution minimising the total cost for the airspace user of the ADCB regulations.	DONE	
TCP-FR-005	The TCP will return a trajectory set with the regulated demand (only delay in current ConOps or delay and/or re-routings in future ConOps).	DONE	See section 2.1.3 and Appendix A for a description of the TCP input-output files.
LI-FR-001	The TCP will query the TP with the list of flights traversing one or several hotspots.	DONE	See section 2.1, Figure 2-1 and section 3.1.3.2.
LI-FR-002	The different alternatives to avoid a hotspot will be feed back to the TCP in a standardised format.	DONE	See section 2.1, Figure 2-1 and section 3.1.3.2. The same format as nominal TP runs (eSo6) is used.
ASP-FR-001	The ASP will receive a 4D trajectories set and the available sector configurations and capacities and it will compute an optimal sector opening scheme following the current ConOps. The sector opening scheme will include for each period of time list of active sectors, including: number of active controllers and traffic load metric per sector. The module will seek for the minimum number of controllers (active sectors) that satisfies the workload limits.	DONE	See section 2.1.2 and Appendix A for a description of the ASP input-output files. See section 3.1.2 and 3.2.2 for example results.
ASP-FR-002	The ASP will provide a functionality for simulating severe weather events on the airspace. For a given Airspace structure, the ASP will introduce the necessary capacity limitations in form of regulations or SAM parametrization. The weather events will have a limited duration.	PARTIALLY DONE	Done for the ASP in static mode ("current ConOps").
ASP-FR-003	The ASP will receive a 4D trajectories set and will design a dynamic sectorization of the airspace, in line with SESAR2020 ConOps (future) based on the	NOT DONE	ASP in SESAR2020 ConOps (Dynamic Airspace Configuration) was not finally integrated in the

	complexity of the received traffic. The airspace dynamic configuration will be provided in terms of SAM groupings for each period of time. This includes a list of active sectors, called Controlled Airspace Block (CAB), not previously defined and built as re-grouping of SAMs (which are defined before the grouping process). The output also contains the traffic load per CAB.		APACHE System (see limitations in section 4.2.2).
RA-FR-001	The RA is considered functionally part of the Performance analyser module, though is separated in the software architecture.	DONE	See Figures 1-2 and 2-1.
RA-FR-002	The RA will receive a set of trajectories to which it will estimate safety PIs and risk of conflict. This set of trajectories could come from the TCP (regulated traffic), TP (planned traffic) or in post-ops assessment from actual trajectories (realised traffic).	DONE	See section 2.1.1 and Appendix A for a description of the PA (which embed the RA) input-output files. All sets of trajectories delivered to the RA use the same file format (eSo6), regardless if they come from the TP, TCP or post-ops data.
RA-FR-003	The RA will detect separation violation between pairwise aircraft.	DONE	See section 3.1.4 and 3.2.4 for example results
RA-FR-004	The RA will compute the minimum separation between pair of aircraft and based on that, conflict severity.	DONE	
RA-FR-005	The RA will compute the duration of separation violations between pairs of aircraft.	DONE	
RA-FR-006	The RA will count traffic alerts, resolution advisories and near mid-air collisions depending on the duration of pairwise separation violations.	DONE	
RA-FR-007	The RA will calculate conflict/accident risks between pairs of aircraft.	DONE	
PA-FR-001	The PA will interface with the TP to process 4D trajectories and summarize information regarding individual flights within scenarios and test cases simulations.	DONE	See section 2.1.1 and Appendix A for a description of the PA input-output files. All sets of trajectories delivered to the PA use the same file format (eSo6), regardless if they come from the TP, TCP or post-ops data.
PA-FR-002	The PA will interface with the TCP to process its outputs and summarize information regarding sets of individual trajectories within scenarios and test cases simulations.	DONE	
PA-FR-003	The PA will interface with the ASP to process airspace sectors outputs and summarize information regarding individual sectorisation configurations within scenarios and test cases simulations.	DONE	See section 2.1.2 and Appendix A for a description of the PA input-output files.
PA-FR-004	The PA will interface with RA to process safety and risk outputs and summarize information regarding individual air traffic patterns within scenarios and test cases simulations.	DONE	The RA is an independent software component that computes safety PIs from input So6 files. From a logic point of view, it is embedded within the PA, which in general terms it the component of the System computing all PIs.
PA-FR-005	The PA will interface with the APACHE database in order to record summarized information related to TP, TCP, ASP and RA.	CHANGED	Finally, a database was not implemented but a consolidated CSV table (see section 2.1.4 and Figure 2-1).
PA-FR-006	The PA will compute the variable denominated *DelayPerFlight*. This variable is the time deviation in arrival of two sets of trajectories with the same flights.	DONE	Variable used for those PIs requiring delay values.

	It will produce as output a vector of arrival delays, where each vector element represents a flight delay. A vector with cancelled flights and a vector of diverted flights will be also produced.		
PA-FR-007	The PA will compute the variable denominated *SectorOccupancyPerHour*. This variable contains three occupancy metrics detailed per airspace sector and per hour. The PA requires as input one 4D trajectory set and an airspace sectorisation scheme. It will produce as output a data structure consisting of number of aircraft, the time spent and the nautical miles flown in each sector per hour.	DONE	Variable used for those PIs requiring sector occupancy values.
PA-FR-008	The PA will compute the variable denominated *EnRouteCharges*. The PA requires as input the list of 4D trajectory sets and the airspace structure (including unit cost charges). The PA will produce as output a vector where each position consists of the ANSP costs (in Euros) per flight.	DONE	Variable used for those PIs requiring en-route charges values.
PA-FR-009	The PA will compute the variable denominated *OpeningSchemeEvaluation*. The PA requires the airspace structure (in term of opening scheme for current ConOps or in terms of SAMs for SESAR2020 ConOps) and a 4D trajectory set. It will output a vector of sector activations and a matrix of sector occupancy. The vector of sector activations contains the number of minutes of each activated sector. The matrix of sector occupancy contains the number of aircraft per active sector and per hour.	DONE	Computed from the *SectorScheme* file provided by the ASP (see Figure 2-1).
PA-FR-010	The PA will have a functionality to estimated burnt fuel for a given flight. The PA will receive a 4D trajectory and the information of the weather. The output of the PA will be the burned fuel in Kg. This functionality can also be depicted as the variable *FuelCalculation*, which will be computed via the PA-TP interface.	DONE	Computed by the PA by using features of DYNAMO (see section 2.1.1.2), which allows to reconstruct post-ops trajectory data (see section 2.1.4).
PA-FR-011	The PA will compute the variable denominated *Transfers*. The PA will receive a 4D trajectory set and an airspace structure. The output will consist on a vector that provides the number of active sectors crossed per flight.	DONE	Computed from the T5 file provided by the ASP (see Figure 2-1).
PA-FR-012	The PA will compute the metrics related to an individual flight. The PA will receive a 4D trajectory and will calculate the total distance flown, the total flight time, the Available Seat Mile (ASM) and the number of flight level changes. The ASM will be computed using a standard number of seats of each aircraft type. This functionality can also be depicted as the variable *EvaluateFlight*, which will be computed via the PA-TP interface.	DONE	Computed by the PA by using features of DYNAMO (see section 2.1.1.2).
PA-FR-013	The PA will compute the variable denominated *CutTrajectorySet_xAU*. This variable has the set of trajectories grouped by airspace user. The PA requires as input one 4D trajectory set. As output the PA will produce a dictionary like structure indexed by airspace user to hold for each one the trajectory set of the flights of the airspace user.	DONE	Variable used for those PIs requiring values separated by Airspace User.

PA-FR-014	The PA will produce graphs for the PIs visualisation. The technology considered in this moment is JavaScript Data-Driven documents. However, the specific technology and the specific graphs to be considered will be selected in a future phase of the software development cycle.	DONE	Besides graphs produced with Excel (from the consolidated CSV file resulting from the PA), more detailed graphs are finally produced with specific Python libraries (same programing language used to code the PA).
PA-FR-015	The PA will calculate the Great Circle Distance between two points in a sphere (given in latitude/longitude coordinates).	DONE	Variable used for those PIs requiring this value.
SCI-FR-001	The TP will format each of the N trajectories, in order to match the input file format of the TCP component. This format consists of longitude, latitude, altitude (meters) and speed (meters per second) per trajectory point discretized in a one-second-time interval.	DONE	An ad-hoc file format (extended So6) has been designed for this purpose (see Appendix C).
SCI-FR-002	The TCP will format flight trajectories in order to match the input file format of the ASP component. This format consists of longitude, latitude, altitude and ground speed per trajectory point, discretized in a one or five -second time interval.	DONE	

Table E-1 Functional requirements

E.2 Non-functional requirements

Software Requirement ID	Description	Fulfilment	Evidences/remarks
TP-NFR-001	The TP will be designed to be the most efficient for the simulation of a flights set. For this purpose, a High-Performance Computing (HPC) approach might be implemented. Specifically, a cluster of computers and a parallelisation prototype for TP could be used.	DONE	The TP has been developed with an HPC approach. See Appendix B.
TP-NFR-002	The TP will be designed to be the most efficient for the simulation of a specific flight. This might include coding optimization techniques for the TP.	DONE	Dynamo uses pre-computed look-up tables to speed-up the optimisation of trajectories and guarantee stability of the algorithm. See (Dalmau et al., 2018) for details.
TP-NFR-003	The TP will be physically located at UPC premises.	DONE	The TP and TCP have been entirely developed by UPC and are is installed in a small cluster of computers (see Appendix B) at UPC premises.
TCP-NFR-001	The TCP will be physically located at UPC premises.	DONE	
LI-NFR-001	The loop interactions between TP and TCP will be designed to be the most efficient possible (HPC).	DONE	
ASP-NFR-001	The ASP will be physically located at ENAC premises.	DONE	The ASP has been entirely developed by ENAC and it is located at their premises. Interfaces with TP, TCP and PA are done by using the NFS shared file system (see section 2.2).
RA-NFR-001	The RA will be designed to be the most efficient for the simulation of an air traffic pattern. For this purpose, an HPC approach might be implemented. Specifically, a cluster of computers and a parallelisation approach.	DISMISSED	After the first integration and scalability tests, an HPC approach was not deemed necessary. Instead, several PCs were used in parallel to run the different Case Studies.

RA-NFR-002	The RA will be physically located at UB-FTTE premises.	DONE	The RA has been entirely developed by UB-FTTE and it is located at their premises. Interfaces with TP, TCP and PA are done by using the NFS shared file system (see section 2.2).
PA-NFR-001	The performance analyser will be physically located at UPC premises.	DONE	The PA has been developed by UPC and UB-FTTE and it is located in the small cluster of computers (see Appendix B) at UPC premises in order to speed-up PA computations.
SCI-NFR-001	All input and output files will be stored in a shared file system located physically at UPC premises.	DONE	A NFS shared file system has been implemented accessible via secure shell. See section 2.2
SCI-NFR-002	The APACHE database will be stored in a shared file system located physically at UPC premises.	CHANGED	Finally, a database was not implemented but a consolidated CSV table (see section 2.1.4 and Figure 2-1) stored in the NFS shared file system mounted in the UPC cluster.
SCI-NFR-003	Each software components will have a human operator in the corresponding partner's premises.	DONE	UPC was responsible to run the TP, TCP and PA; ENAC responsible to run the ASP and UB-FTTE to run the RA component.
SCI-NFR-004	The TP and the TCP components will process input files and write output files directly over the shared file system	DONE	The TP and TCP have been installed in a small cluster of computers (see Appendix B) at UPC premises, in which the NFS shared file system described in section 2.2. is mounted.
SCI-NFR-005	The ASP and the RA component will not process input files and write output files directly over the shared file system. In this sense, inputs file shall be copied to local storage at each partner premises and the output files copied to the shared file system.	CHANGED	Finally, the NFS shared file system (see section 2.2) was available via secure shell to all partners, who uploaded/downloaded the necessary files (see Figure 2-1).

Table E-2 Non-functional requirements

E.3 Domain requirements

Software Requirement ID	Description	Fulfilment	Evidences/remarks
TP-DR-001	Flight origin and destination airport for simulation test cases will be obtained from DDR2.	DONE	Data obtained from DDR2's exp2/So6 files (see section 2.1.1).
TP-DR-002	Flight ETD and ETA for simulation test cases will be obtained from DDR2.	DONE	
TP-DR-003	Flights aircraft type for simulation test cases will be obtained from DDR2.	DONE	
TP-DR-004	Aircraft performance models will be obtained from BADA 4.x.	DONE	BADA 4.1 licence granted to UPC for the purposes of the APACHE project.
TP-DR-005	Weather information will be obtained via GRIB2 files from NOAA or ECMWF.	DONE	GRIB2 files processed as input of the TP (see 2.1.1.2).

ASP-DR-001	FABs definition will be obtained from DDR2 or FAB dedicated documentations available at official websites.	DONE	Data obtained from DDR2 and alternative and complementary sources (see 2.1.2).
ASP-DR-002	Airspace blocks definition will be obtained from DDR2 or EAD.	DONE	
ASP-DR-003	ESs definition will be obtained from DDR2 or EAD.	DONE	
ASP-DR-004	CSs definition will be obtained from DDR2 or EAD.	DONE	
ASP-DR-005	CONFs definition will be obtained from DDR2 or ACC internal documentation.	DONE	
ASP-DR-006	Airspace sector capacities will be obtained from ACC internal documentation.	DONE	
RA-DR-001	Uncertainties about flights sector entry time and flight velocities (necessary for simulation) in the form of probability density functions will be assumed based on expert judgement.	NOT DONE	The consideration of uncertainty is finally out of the scope of the APACHE simulations
PA-DR-001	See TP-DR-005 (Weather)	DONE	
PA-DR-002	Flight arrival delay cost and flight cancelation/diversion costs will be modelled using existing state-of-the-art bibliography. This model is referred as *DelayCostModel.*	DONE	Cost of delay taken from (Eurocontrol, 2015)
PA-DR-003	Airline data shall be estimated. This includes cost-indexes and payload weights for each aircraft of the airline.	NOT DONE	Not fully implemented. Cost Index estimated for post-ops assessments, but not Payload (educated guess assumed).
PA-DR-004	The Available Seat Mile (ASM) information (number of seats per aircraft type) shall be estimated from a public/private data base	DISMISSED	Data finally not used by the PA.
PA-DR-005	Radar data for the actual flights (for scenario recreation or historical scenario assessment) will be obtained from ANSPs or using DDR2 M3 files if ANSP data is not available.	DONE	Data obtained from DDR2 and also PRU (correlated position reports).
PA-DR-006	Data about the planned flights (for scenario recreation or historical scenario assessment) will be obtained from ANSPs or using DDR2 M1 files if ANSP data is not available.	DONE	Data obtained from DDR2.
PA-DR-007	The PA will convert from kg of fuel to euros using some external source (to be identified) or assuming an input value.	DONE	Price of fuel taken from (Eurocontrol, 2015)
PA-DR-008	The PA will convert from kg of fuel to kg of CO2 or other emissions using some external source (to be identified) or assuming some basic conversions.	DISMISSED	Emissions finally not computed by the PA.
PA-DR-009	Regulated flight plans (for scenario recreation or historical scenario assessment) will be obtained from ANSPs or using DDR2 M2 files if ANSP data is not available.	DONE	Data obtained from DDR2.
PA-DR-010	Current airspace structure including opening scheme and sector capacities will be obtained from DDR2.	DONE	Data obtained from DDR2 and alternative and complementary sources (see 2.1.2).

Table E-3 Domain requirements

Appendix F Validation of the APACHE Trajectory Planner

Two different validation exercises have been done:

- comparison with trajectories generated with the **Airbus performance engineering programs** (PEP); and
- comparison with trajectories generated by the **AURORA SESAR Exploratory Research Project**.

PEP is an application designed to provide flight performance engineers with the necessary tools to handle the performance aspects of flight preparation, and also to analyse aircraft performance after the flight. The Airbus PEP comprises several modules. The flight planning module (**FLIP**) allows to produce fuel predictions for a given flight under simplified meteorological conditions (i.e. constant wind), accounting also for airline cost policies and aircraft performance capabilities.

AURORA is also a SESAR Exploratory Research (ER) Project in the topic of ATM performance. As consequence of the coordination activities among all SESAR ER projects in the same topic, a cross-validation activity was proposed in order to compare the trajectories obtained by the APACHE TP with those obtained by the homologous tool used in AURORA, which is developed by Boeing Research and Technology Europe (BRTE).

F.1 Validation with Airbus PEP software

The validation exercise has been performed by means of **18 Test Cases (TC)**. For each TC, a trajectory computed by the APACHE TP has been compared with that provided by PEP FLIP module using the same input parameters. The input parameters include the **aircraft model**, the **landing mass**, the **CI** and the **ground distance** from origin to destination. The set of Test Cases is obtained as a result of combining different values of these input parameters.

F.1.1 Description of the test cases and statistical indicators used

Three representative wide-body Airbus models, which are included in the PEP database, have been used for the validation exercise: The Airbus A320-213, a short to medium-range aircraft comparable to the Boeing B737; the Airbus A330-321, a medium to long range aircraft; and the Airbus A340-231, a long-range aircraft comparable to the Boeing B777.

For each aircraft model, trajectories for **3 landing masses** and **2 CI** have been investigated. On the one hand, the following landing masses have been considered, which are expressed as a percentage of the Maximum Landing Mass (MLM): 75%, 90% and 100%. On the other hand, the trajectory that minimises fuel consumption (i.e. CI = 0 or maximum range), and that using a CI representative of long range operations have been analysed. The CI equivalent to long range operations for each aircraft model has been obtained from a database bundled in the APACHE TP. The CI values in this study are 45, 98 and 400 kg min^{-1} for the A320, the A330 and the A340 models, respectively.

Finally, a **unique (and representative enough) ground distance** has been selected for each aircraft model. These distances are, respectively, 1500, 2500 and 3000 NM for the A320, A330 and A340 models. It is important to remark that all TCs have been performed assuming a hypothetical straight-line route from origin to destination, in International Standard Atmosphere (ISA) conditions, and without considering winds. In addition, trajectories were computed assuming a standard flight level allocation and orientation scheme.

In this validation exercise, **mean** and **maximum absolute relative** fuel consumption and flight time figures have been considered as indicators for the comparison between trajectories generated by the FLIP module of PEP and the APACHE TP. Differences in optimal speed and altitude profiles have also been investigated.

Optimal speed differences for a given TC are computed by comparing the optimal Mach number for the longest cruise phase of the APACHE TP and PEP trajectory.

Optimal altitude profile differences for a given TC are visually inspected by computing the distance devoted in climb and descent, and the flight level and distance spent in the cruise level (or levels if one or more step climbs are performed).

F.1.2 Optimal speeds and Altitudes

Figure F-1 compares, for each TC, the optimal cruise Mach[14] for the trajectories generated by the APACHE TP and PEP.

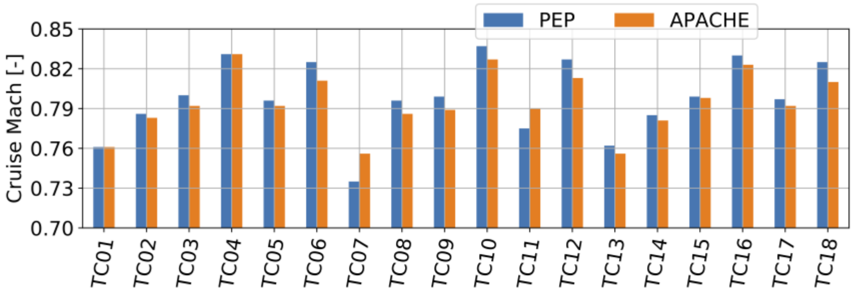


Figure F-1 Optimal cruise Mach number comparison

As expected, the optimal cruise Mach number increases with the CI and slightly decreases with the landing mass, for fixed values of the remaining input parameters.

According to Fig. F-1, the optimal cruise Mach for both APACHE TP and PEP trajectories is very similar, independently of the input parameters. It can be also observed that the optimal cruise Mach of APACHE TP is slightly lower than that of PEP for most of the TCs. This is probably caused by the differences in the aircraft performance model used by the trajectory optimisation tools compared herein: The APACHE TP implements the Base of Aircraft Data (BADA) v4.x, while PEP uses tabulated performance data, directly obtained from flight tests, and with a high level of accuracy.

It is also worth noting that the higher differences in optimal cruise Mach are observed for test cases from TC07 to TC12 (inclusive) which correspond to those trajectories computed with 75% of the MLM. The main reason of such larger differences is the fact that **PEP restricts the value of the maximum allowed flight level**, while the APACHE TP allows to climb at a higher altitude provided that the minimum rate of climb restriction is satisfied (i.e. the aircraft is able to climb with a rate of climb higher

---

[14] It should be noted that the optimal speed (in ISA conditions and calm winds) only depends on the mass of the aircraft and the altitude. Since a trajectory might have different cruise altitudes (aircraft might progressively climb as fuel is burnt), different optimal cruise Mach might be also observed for the same trajectory. In this validation exercise, the optimal cruise Mach number for the longest cruise flight level has been taken for the comparison.

or equal than **500** ft min^{-1}) and that the next altitude is better than the actual one, in terms of specific cost (i.e. flight cost per unit distance).

This limitation in the maximum allowed cruise level can be better appreciated in Fig. F-2, which compares the altitude profile of the trajectory computed by the APACHE TP (dashed) with that of PEP (solid) for each TC. Each flight is composed by several coloured blocks, corresponding to the different flight phases.

The first block (red) shows the total ground distance spent in the climb (CMB) phase, from take-off to the top of climb; the last block (cyan) shows the total ground distance dedicated to the descent phase (DES), from the top of descent to landing. In between these two blocks, up to three blocks could appear in the plot, each one showing the ground distance flown in a given cruise level (CRZ LVL). In addition, the blocks representing a CRZ LVL include a label specifying the flight level associated to that phase. The short climbs between two consecutive flight levels are not displayed in this figure.

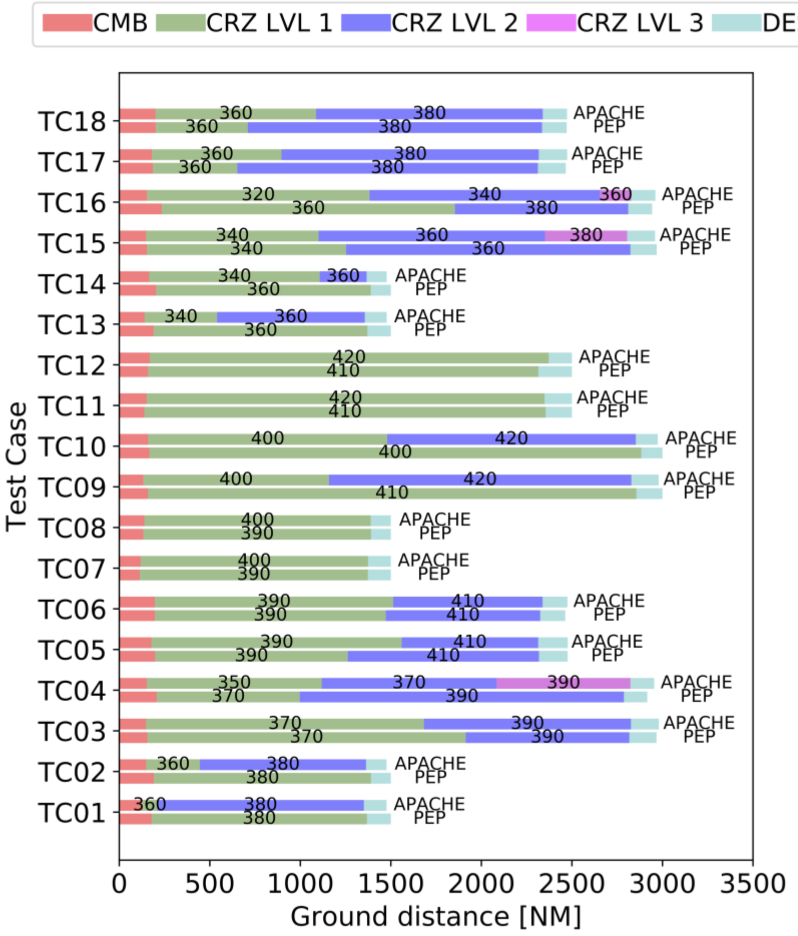


Figure F-2 Optimal altitude profile comparison

As seen in Fig. F-2, the trajectories computed either with the APACHE TP or the PEP FLIP, select the same or very close cruise flight levels. As commented before, the higher differences are found for those TC in which trajectories were computed with a landing mass corresponding to 75% of MLM, where a maximum altitude bound is reached in PEP. In addition, the most similar trajectories, in terms of altitude profile, are found for the A330; and the most different trajectories are those computed using the A320 and A340 models in long range operations, independently of the landing mass.

It can be also observed that the APACHE TP generates trajectories with the same number of step climbs than those computed by PEP, or with one more in some cases.

It is also worth noting that, as expected, the distance devoted to climb (resp. descent) increases (resp. decreases) with the CI, for a given landing mass, aircraft model and ground distance from origin to destination. In other words, both the top of climb and top of descent "move" forwards with an increasing CI (all the other input parameters fixed).

F.1.3 Fuel consumption and flight time

Figure F-3 shows, for each TC, a graphical comparison of the fuel consumption and flight time figures, resulting from the PEP and APACHE TP trajectories. As expected, both APACHE TP and PEP show larger fuel consumption and smaller flight times figures as the CI increases, fixing the remaining input parameters. In addition, for a given combination of ground distance from origin to destination, aircraft type and CI, the fuel required increases with the landing mass.

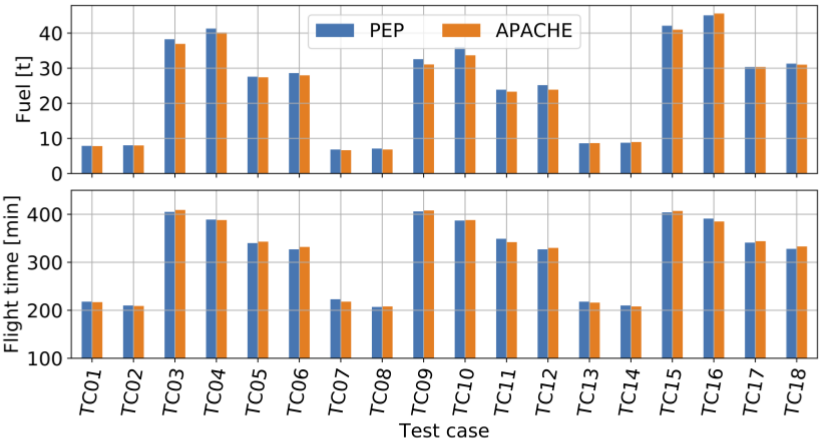


Figure F-3 Fuel consumption and flight time comparison

According to Fig. F-3, the trajectories generated by the APACHE TP and PEP are comparable in terms of flight time and fuel consumption. In most of the cases, the APACHE TP reported slightly less fuel consumption. The relative mean absolute difference in fuel consumption is around **2.4%,** being **6.6%** the relative maximum absolute difference (TC10). Regarding the flight time, it is more difficult to take general conclusions and for some TC, the APACHE trajectories experience longer trip times, while for other TC the PEP trajectories are slower. This behaviour can be explained by the altitude differences in the cruise phase (see Fig. F-2). Nevertheless, the differences remain very small, with a relative mean absolute difference of **1.0%** and a relative maximum absolute difference of **2.2%,** (TC07).

It can also be observed that, in general, those flights with more differences in the altitude profile and optimal cruise Mach show also larger differences in flight time and fuel consumption figures. In addition, the most significant differences are observed for those TCs in which a landing mass corresponding to 75% of the MLM was used.

F.1.4 Conclusions

Results arising from this validation exercise show that trajectories generated by the APACHE trajectory predictor (TP) are comparable to those obtained from PEP flight planning module for the same input parameters, in terms of fuel consumption and flight time figures.

In addition, the optimal cruise speeds and altitude profiles are very similar, proving that the traffic patterns obtained with the APACHE TP will accurately represent current operations. However, it should be noted that the validation exercise presented in this document is scoped to only Airbus models, and that the effects of realistic non-standard atmospheric conditions and wind fields have been not analyse, because of the limitations of the PEP regarding the modelling of realistic meteorological conditions. As a concluding remark, this validation exercise only applies for the performance models and optimisations algorithms implemented in the vertical profile optimisation module of the APACHE TP.

F.2 Validation with AURORA Trajectory Predictor

The validation exercise has been performed for **1583 input flights**, which represent a statistically meaningful sample of the traffic that overflew **France** during the **20th of February 2017**. These traffic data have been obtained from the Demand Data Repository (DDR2) of EUROCONTROL, which allows to download historical traffic data containing filed flight plan data (M1 files), as well as regulated flights (M2 files), or actual trajectories (M3 files). The aircraft type (e.g. A320), scheduled departure time, and origin and destination airports for each one of the 1583 flights were extracted from the M1 and used as inputs for both APACHE and AURORA TPs.

An optimal trajectory, however, was not obtained for all these flights. In some circumstances, either the APACHE or the AURORA TP was not able to find a solution, due to convergence issues or inappropriate inputs. Consequently, the number of flights from which statistics will be generated is, at most, 1583; being the number of failed flights dependant on the optimisation objective and constraints of each particular Test Case.

In order to be consistent, the weather data (including wind, temperature and pressure) for the same day and region of study have been considered when optimising the trajectories. Yet, each TP has gathered and processes this data independently.

Both APACHE and AURORA trajectory planners generated a representative set of optimal trajectories under the same conditions: using the same traffic demand (origin-destination pairs, aircraft type and departure day/times), objective function (which includes the CI), en-route ATM constraints and aircraft payload.

F.2.1 Description of the test cases and statistical indicators used

This validation exercise is composed by three well differentiated Test Cases, in which both APACHE and AURORA TPs were configured to optimise the trajectories using the same optimisation objective and/or ATM en-route constraints.

In the **first test case**, the trajectories were computed by **fixing the lateral routes** to those initially planned by the airspace user, and optimising the vertical profile in a full Continuous Cruise Climb (**CCC**) scenario, with all the flights minimising their operating costs according to a **representative CI** of current operations. The planned route was obtained from the M1 file (filed flight plan).

In the **second test case**, the trajectories were computed by optimising the lateral route in a **hypothetical full free route airspace** (from origin to destination airports), but still considering weather effects (thus aiming to follow most favourable wind and temperature profiles). The vertical profile was optimised assuming **CCC** operations and considering a **representative CI** of current operations (for each flight, the same CI as used in Test Case 1).

Finally, in the **third test case**, the trajectories were computed by optimising the lateral route in a **hypothetical full free route airspace** (from origin to destination airports), as done in Test Case 2; but computing the vertical profile assuming **CCC** and minimising only fuel consumption (i.e. **CI=0** or maximum range).

Different statistical indicators can be used to assess the similarity between two data sets. In this validation exercise, the following indicators have been selected:

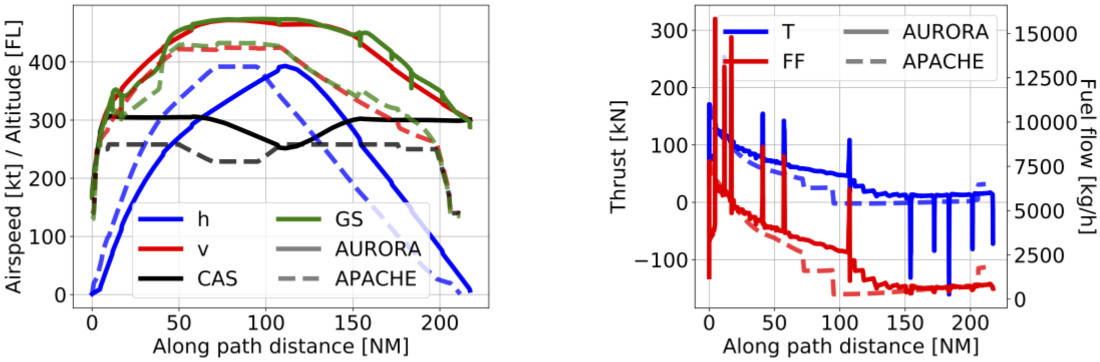
- the mean ($\bar{\Delta}$) and median ($\tilde{\Delta}$) difference between APACHE and AURORA key trajectory figures, such as fuel consumption, trip time or trip distance;
- the Interquartile Range (IQR) of this difference, which measures the statistical dispersion and it is equal to the difference between the 75th and the 25th percentiles;
- the correlation coefficient (ρ_(X,Y)), which measures how strong a relationship between two variables X and Y is; and
- the Kolmogorov–Smirnov parameter (*K-Test*), which is a nonparametric test of the equality of continuous, one-dimensional probability distributions that can be used to compare two samples. This test is typically used to determine whether the two samples are likely drawn from the same distribution or not. If the p-value resulting from the K-Test is greater than 0.05, the hypothesis that the distributions of the two samples are the same cannot be rejected. Otherwise, the distribution of the two samples subject of analysis can be considered statistically different.

For the three test cases, **trip time**, **fuel consumption** and **trip (ground) distance** values, along with **cruise altitude** and **cruise speed** differences are compared between the optimal trajectories computed by AURORA TP and those generated by the APACHE TP. Since only CCC operations are compared in this study, it is not straightforward to determine which segments of the trajectory correspond to the cruise and which ones to the climb or descent. As a first approximation, a *virtual cruise altitude* (and *speed*) for each flight is computed as the **median** of the altitudes (and Mach numbers) above **FL300**.

F.2.2 Initial comparison at trajectory level

Before presenting the comparison results at aggregate level, Figures F-4 to F-6 compare the vertical and speed profiles computed by the AURORA trajectory predictor with those obtained with the APACHE tool, for three different trajectories and with increasing flight distances. They also compare aircraft thrust and fuel flow along the trajectory.

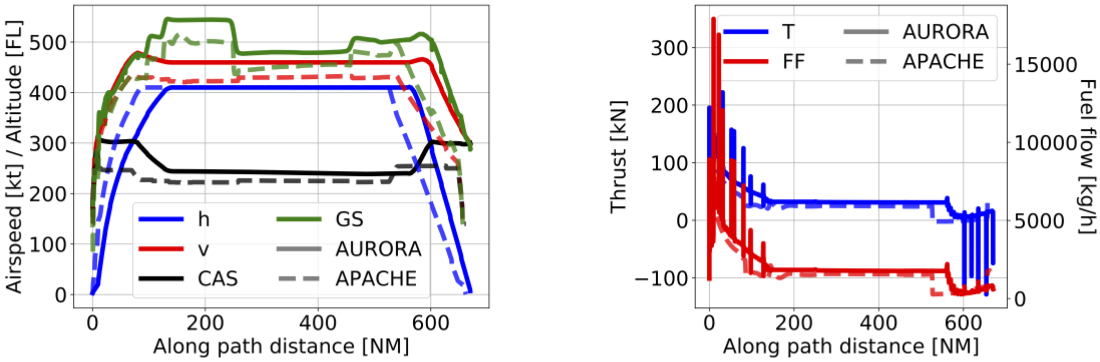
As seen in Figures F-4 to F-6, for these particular flights the cruise altitudes selected by the APACHE and AURORA TPs are very similar. Yet, the climb phase of the APACHE trajectories is always steeper, reaching the top of climb much earlier, even if the thrust force is lower. This apparent paradox could be explained by the slower speed profiles observed in the APACHE trajectories, which in turn lead to a smaller drag force, more excess thrust and, finally, a higher climb gradient.



(a) Vertical and speed profiles

(b) Thrust and fuel flow

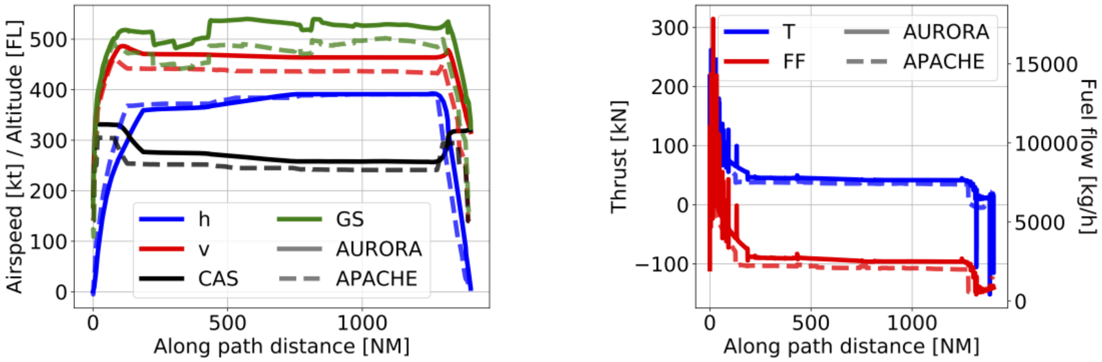
Figure F-4 Detailed comparison for example flight #1



(a) Vertical and speed profiles

(b) Thrust and fuel flow

Figure F-5 Detailed comparison for example flight #2



(a) Vertical and speed profiles

(b) Thrust and fuel flow

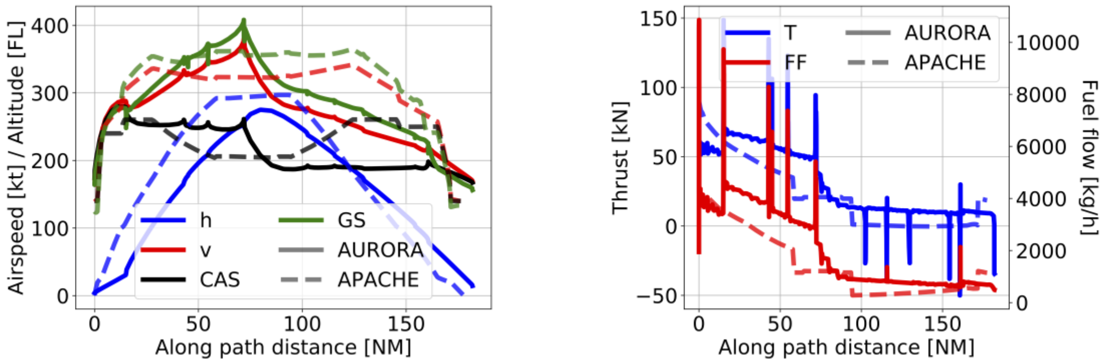
Figure F-6 Detailed comparison for example flight #3

These differences in speed are caused by the **discrepancies in the drag model** used by the two TPs compared herein. On the one hand, the AURORA TP implements the Base of Aircraft Data (BADA) **v3.x** aircraft performance model, which neglects the air compressibility effects that appear at high speeds. On the other hand, the APACHE TP implements **BADA v4.2**, which models the dependences of the drag coefficient with the Mach number, resulting in slower optimal speeds for the same flight conditions and CI.

Moreover, it can be noticed from Figures F-4 to F-6 (b) that the AURORA thrust and fuel flow show regular spikes, which can be also observed to a less extent in the ground speed profiles. These spikes could be caused by the discretisation scheme used by the AURORA TP.

It is also worth noting that, for short flights, the **CCC are inappreciable**, and optimal profiles fluctuate around a given altitude (yet not forced to be a round flight level) with more favourable winds and/or temperature values. CCC are easily observed for longer flights, where the mass loss due to the fuel burnt becomes more relevant.

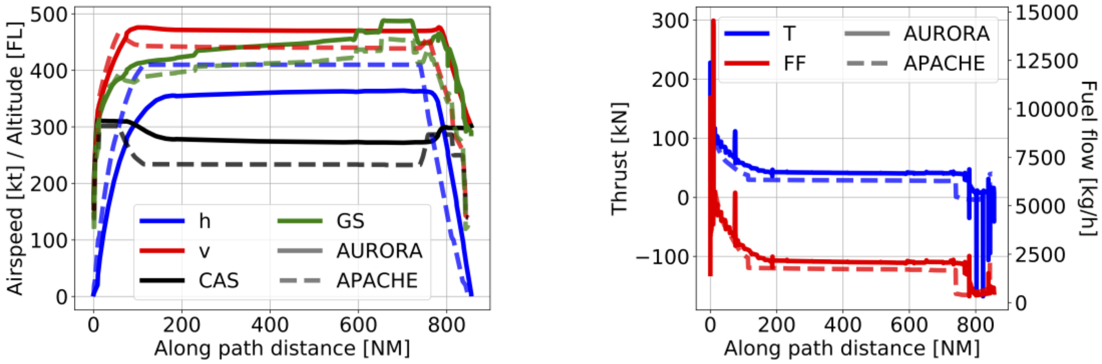
In contrast to previous figures, Figures F-7 and F-8 show examples of flights in which the altitudes selected by the APACHE and AURORA TPs differ considerably.



(a) Vertical and speed profiles

(b) Thrust and fuel flow

Figure F-7 Detailed comparison for example flight #4



(a) Vertical and speed profiles

(b) Thrust and fuel flow

Figure F-8 Detailed comparison for example flight #5

The significant differences found in altitude and speed profiles lead to noticeable discrepancies in total fuel consumption and flight time figures. In particular, for flight #4 the fuel consumption and time differences are around 365 kg (29%) and 5 min (12%), respectively; while for flight #5 these values are about 330 kg (7%) and 11 min (9%), respectively. In both cases, the APACHE trajectory leads to less fuel consumption and more flight time.

An aggregated analysis of the cruise altitude, cruise speed, flight time, ground distance and fuel consumption differences is presented in next sections, using the complete set of flights considered in this validation exercise.

Last but not least, the detailed comparison performed in this section proves that both APACHE and AURORA TPs make use of **the same weather data**, being similar the difference between the ground and true speed.

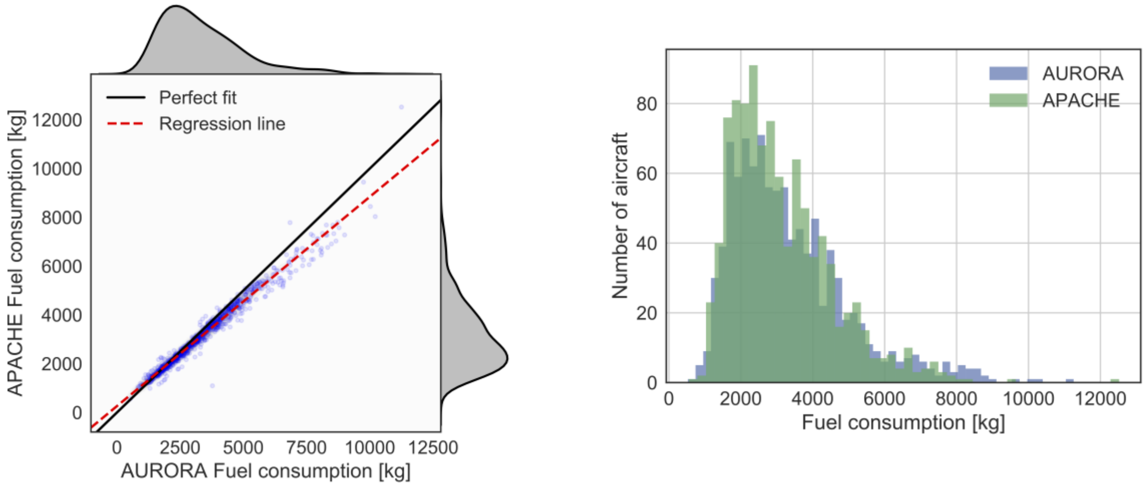
F.2.3 Results for Test Case 1 (Structured routes with typical CI)

In this Test Case, the difference in trip distances between AURORA and APACHE flights is null, since both TP used exactly the same route (taken from M1). Consequently, only flight time and fuel consumption differences will be observed, as a result of mismatches in the vertical and speed profiles.

Figure F-9 compares the **fuel** consumption distribution of APACHE and AURORA trajectories for this Test Case. In Fig F-9a, each point represents a particular flight, the red-dashed line is the linear regression that better fits the experimental cloud of points, and the black-solid line represents the perfect fit (i.e. obtaining exactly the same result from AURORA and from APACHE trajectory predictors). Points above (resp. below) this black line are associated with flights in which the APACHE optimal trajectory lead to more (resp. less) fuel consumption than the corresponding fuel computed by AURORA.

As seen in Fig F-9b, the fuel consumption distributions of APACHE and AURORA are quite similar. In APACHE, however, more flights can be observed in the interval 1,500-2,500 kg, and AURORA results show a larger number of flights above 8,000 kg. Fig F-9a, shows that, for small fuel consumption flights, APACHE and AURORA trajectories result in very similar figures, but the excess fuel consumption of AURORA trajectories with respect to those of APACHE increases with the fuel consumption.

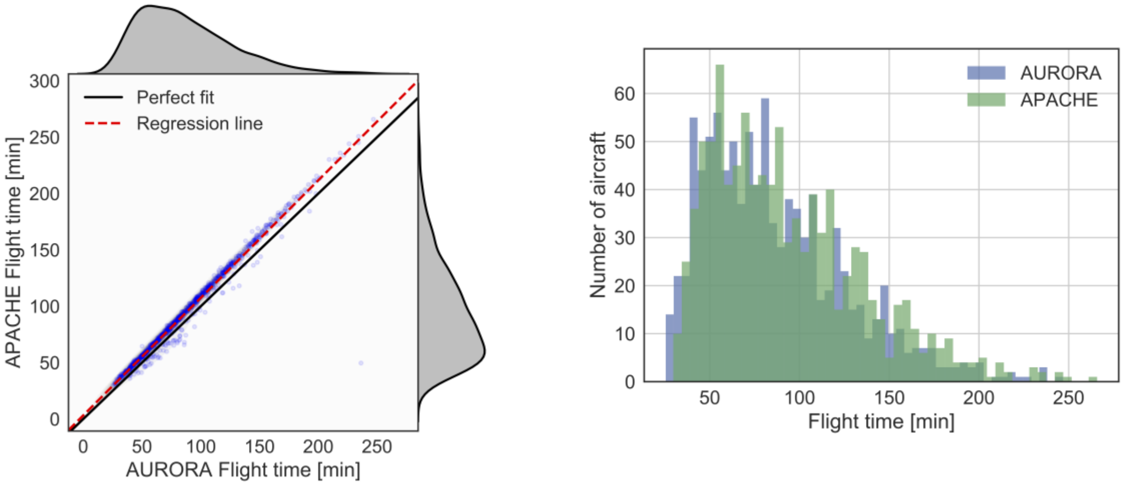
The analogous results, when comparing trip time, are presented in Figure F-10.



(a) Scatter plot

(b) Histogram

Figure F-9 Fuel consumption comparison for Test Case 1



(a) Scatter plot

(b) Histogram

Figure F-10 Trip time comparison for Test Case 1

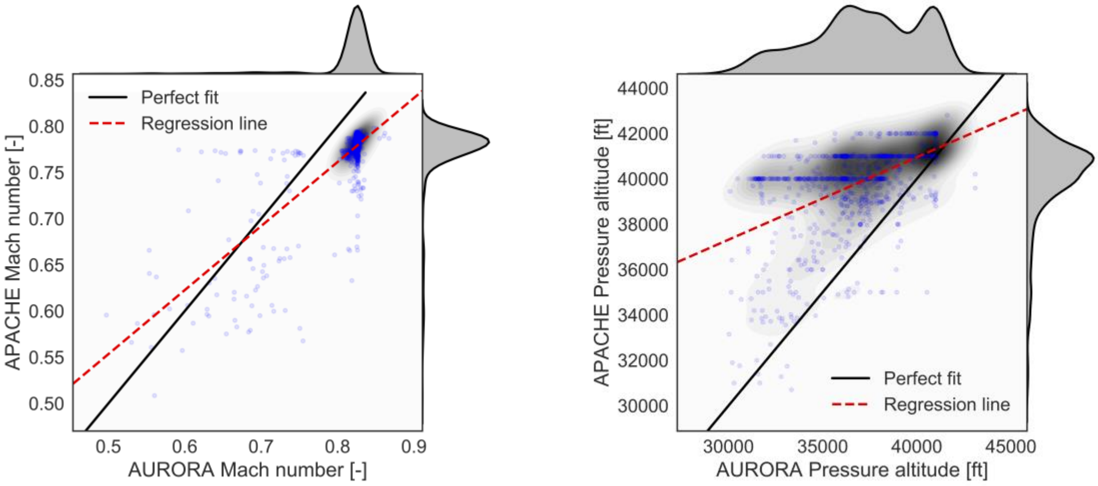
The distributions of flights, as a function of the trip time, for the traffic scenarios computed by the two trajectory predictors are also very similar. Nevertheless, the APACHE distribution is centred to a higher flight time if compared to that of AURORA. This fact can be better appreciated in Fig F-10a, where the trajectories of APACHE clearly show a larger flight time. Results agree with those found for the fuel consumption, since lower fuel consumption are typically associated with larger flight times. The lower flight times observed for the APACHE trajectories are a direct consequence of the slower speed profiles chosen for the APACHE trajectory predictor if compared to that of AURORA, as discussed in Section F.2.2.

In Figures F-9 and F-10 some outliers can be easily observed, which correspond to flights with **large trip distance differences** between the trajectories computed by APACHE and AURORA TPs, **even if both are supposed to fix the lateral route to that initially planned by the airspace user (M1 file)**. A first look into these particular flights revealed that the route of the AURORA trajectory was not the same as that obtained from the M1.

Figure D-11 compares the cruise Mach and altitude above FL300 for this particular Test Case (computed as the median of Mach numbers and altitudes above FL300).

According to Fig. F-11a, the optimal cruise Mach clearly shows a Gaussian distribution, cantered at 0.78 for APACHE and 0.81 for AURORA. It can also be observed that the cruise Mach variance is slightly lager in APACHE. Results shown in this Figure agree with those found in Fig. F-10, in which larger trip times (slower speeds) were observed for the APACHE trajectories.

Regarding the cruise altitude, Fig. F-11b shows that, generally speaking, the APACHE trajectories fly at higher cruise altitudes. Another conclusion arising from this Figure is that the dispersion of cruise altitudes is much higher in AURORA than in APACHE. This assertion can be observed in most of the flights shown in Section F.2.2, where for the APACHE trajectories the cruise level is reached much earlier, leading to a higher median value for the altitudes above FL300 if compared with AURORA.



(a) Cruise Mach (median) (b) Cruise altitude (median)

Figure F-11 Cruise Mach and Altitude comparison for Test Case 1

Wrapping up, AURORA trajectories show higher fuel consumption figures and smaller flight times. It is also worth noting that the dispersion of fuel consumption and flight time differences is noticeable, with IQRs of 260 kg and 4.5 min, respectively. Results from the K-Test for the fuel consumption show a p-value close to 0.05, while for the flight time it can be inferred that the AURORA and APACHE samples are draw from different distributions. Finally, both flight time and fuel consumption correlation coefficients show strong uphill (positive) linear relationship. Conversely, for the cruise Mach and altitude, the p-value obtained from the K-test illustrates that the cruise Mach and altitudes are drawn from different distributions. Moreover, the mean and median differences in cruise Mach and altitude are not negligible.

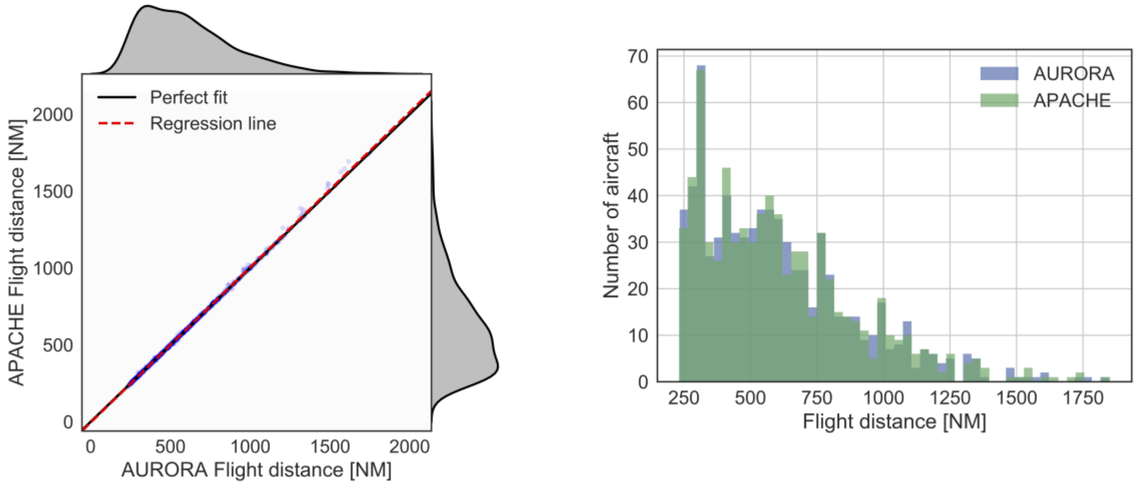
F.2.4 Results for Test Case 2 (Full free route with typical CI)

Figure F-12 shows the comparison of the trip distance distributions between the APACHE and AURORA trajectories. As seen in Fig. F-12a an almost **perfect relationship between the computed trip distances** is found, with most of the scattered points lying in the perfect fit line. It should be noted that the distance shown is the ground distance, meaning that **both trajectory optimisers found the same optimal route, based on the input weather conditions.**

Figures F-13 and F-14 compare, respectively, the fuel consumption and trip time distributions of APACHE and AURORA trajectories; Figure F-15 compares the cruise Mach and altitude between the APACHE and AURORA trajectories.

As expected, the differences in trip distance are insignificant, with both mean and median difference lower than 1NM and an IQR lower than 5NM. In addition, the AURORA trajectories show significantly higher fuel consumption figures and smaller flight times, as observed in Test Case 1. It is also worth noting that the dispersion of fuel consumption and flight time differences is noticeable, with IQRs of 236 kg and 4.8 min, respectively. Results from the K-Test for the flight time and flight distance show that the APACHE and AURORA samples are likely drawn from the same distribution, while for the fuel consumption this hypothesis can be rejected. Finally, both correlation coefficients show strong uphill (positive) linear relationship. Regarding the cruise Mach and altitude, results from the K-test show that both cruise Mach and altitude are likely drawn from different distributions. As in Test Case 1, the mean and median cruise Mach and altitude differences cannot be neglected.

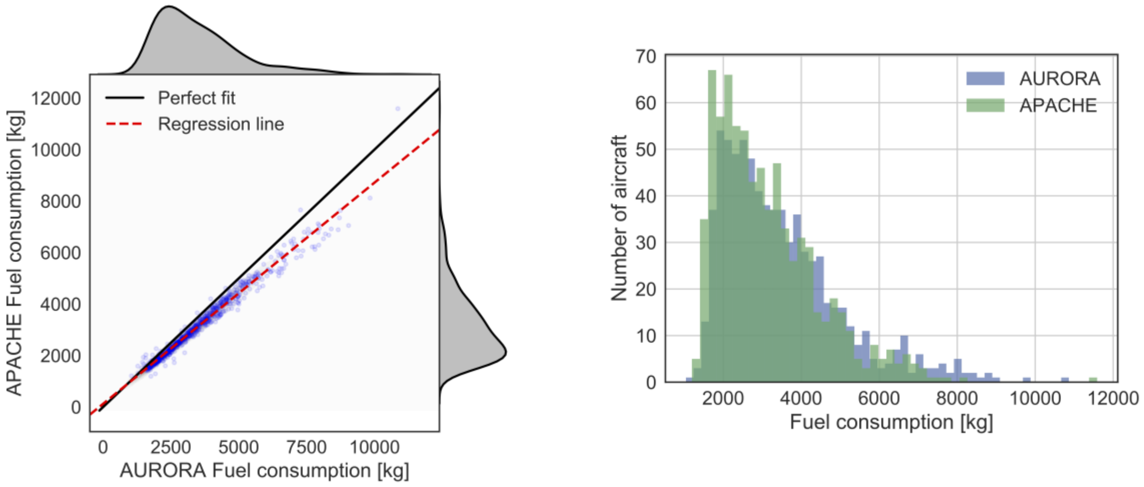
Summing up, flight time, fuel consumption, cruise Mach and cruise altitude results for this Test Case are very similar to those observed in Test Case 1. In both experiments, the lateral routes of APACHE and AURORA were almost identical: in Test Case 1 because the route was fixed to that initially planned by the airspace user; in Test Case 2 because the lateral route optimisation in a free route airspace leaded to very similar flight distances, as shown in Fig. F-12. Therefore, the differences in flight time, fuel consumption, cruise Mach and cruise altitude are only due to differences in the vertical optimisation, as in Test Case 1.



(a) Scatter plot

(b) Histogram

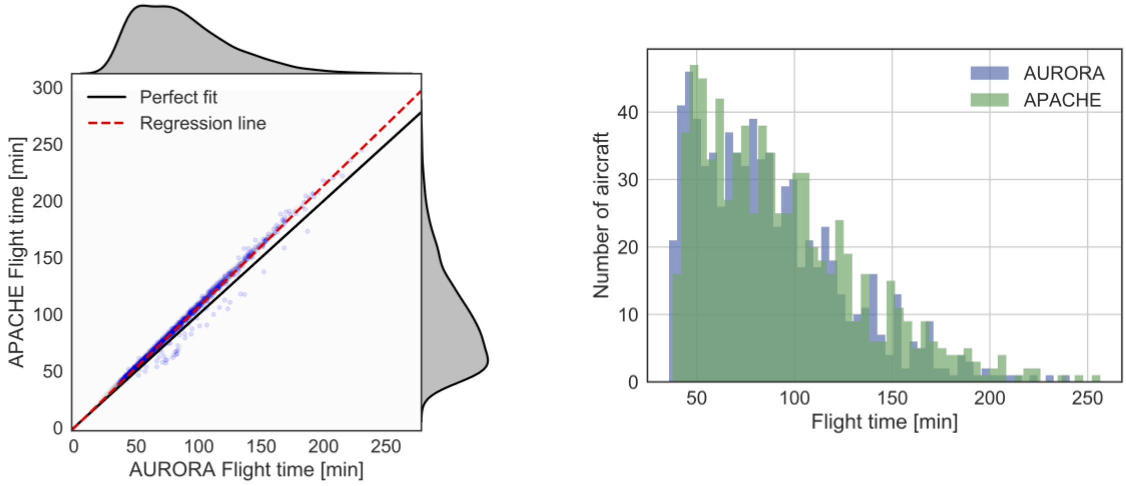
Figure F-12 Trip distance comparison for Test Case 2



(a) Scatter plot

(b) Histogram

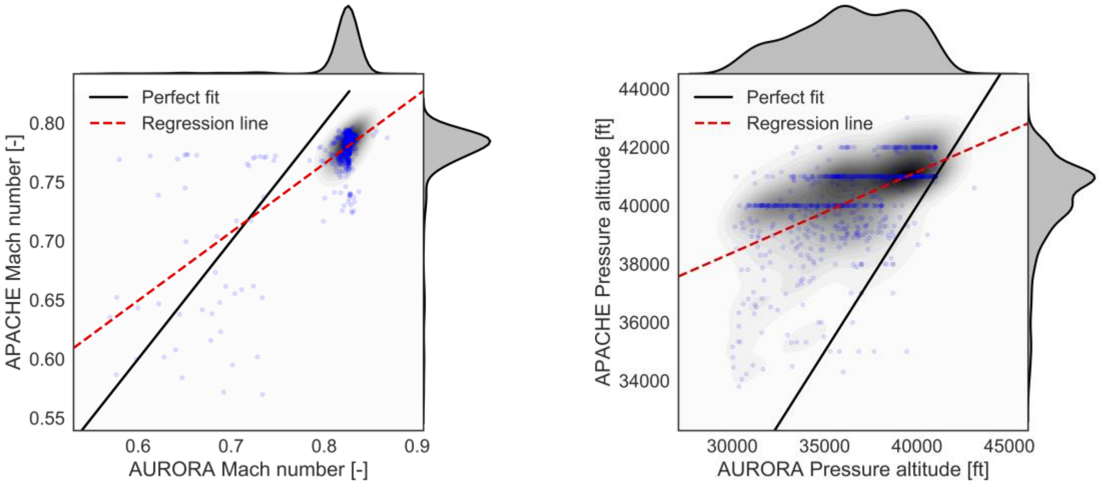
Figure F-13 Fuel consumption comparison for Test Case 2



(a) Scatter plot

(b) Histogram

Figure F-14 Trip time comparison for Test Case 2



(a) Cruise Mach (median)

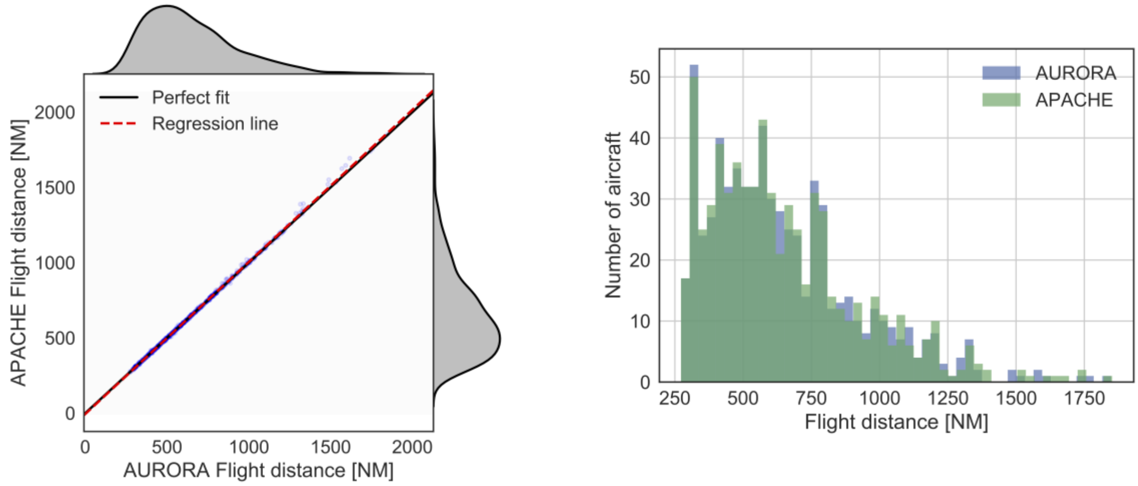
(b) Cruise altitude (median)

Figure F-15 Cruise Mach and Altitude comparison for Test Case 2

F.2.5 Results for Test Case 3 (Full free route with CI=0)

Figure F-16 shows the trip distance distributions for the APACHE and AURORA trajectories, while Figs. F-17 and F-18 show, respectively, the fuel consumption and trip time distributions. Fig. D-19 compares the cruise Mach and cruise altitude distributions.

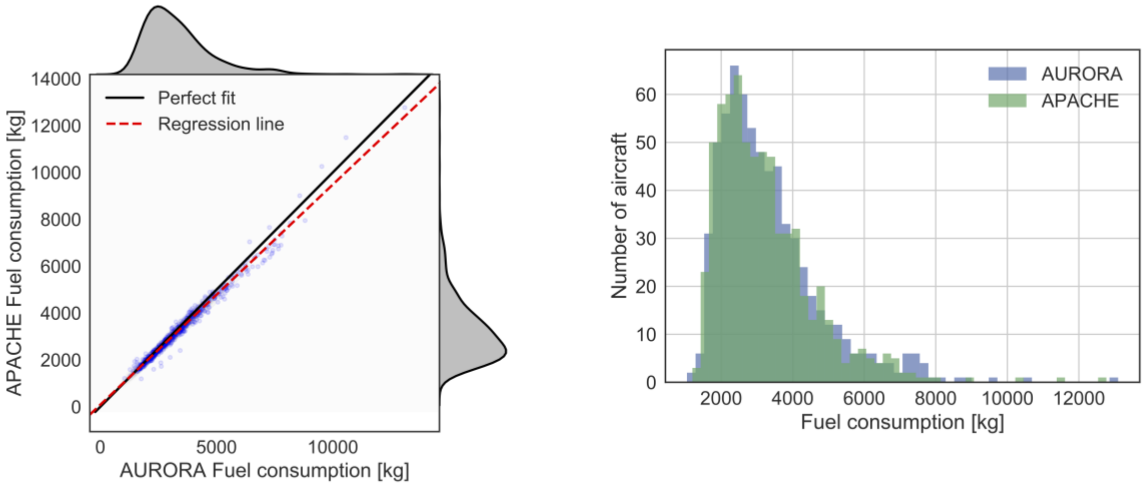
Figure F-17 shows similar results to those found in Figure F-13: for small fuel consumption APACHE and AURORA trajectories result in very similar figures, but the excess fuel consumption of AURORA trajectories with respect to those of APACHE increases with the fuel consumption. Surprisingly, for maximum range operations APACHE trajectories burn similar or less fuel consumption than those of AURORA in the same conditions and also achieve shorter flight times. This apparent inconsistency is probably caused by mismatches in the aircraft performance models used by APACHE and AURORA trajectory predictors.



(a) Scatter plot

(b) Histogram

Figure F-16 Trip distance comparison for Test Case 3



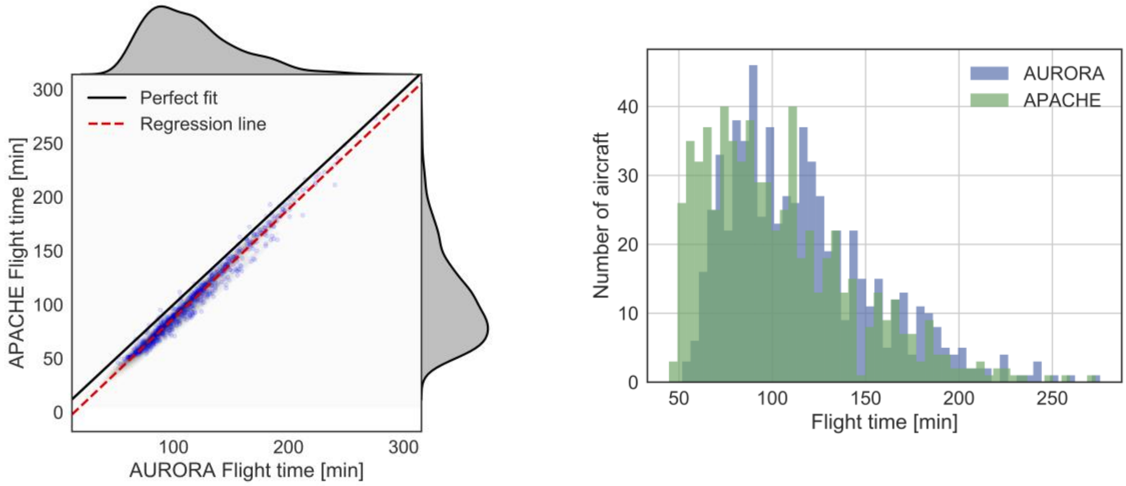
(a) Scatter plot

(b) Histogram

Figure F-17 Fuel consumption comparison for Test Case 3

As expected, the differences in flight distance are insignificant, with a mean and median difference of less than 1NM and an IQR lower than 5NM.In addition, the AURORA trajectories show significantly higher fuel consumption and flight time figures. It is also worth noting that the dispersion of fuel consumption and flight time differences is noticeable, with IQRs of 194 kg and 6.6 min, respectively. Results from the K-Test for the flight distance and fuel consumption show that the APACHE and AURORA samples are likely drawn from the same distribution, while for the flight time this hypothesis can be rejected. Finally, both correlation coefficients show strong uphill (positive) linear relationship.

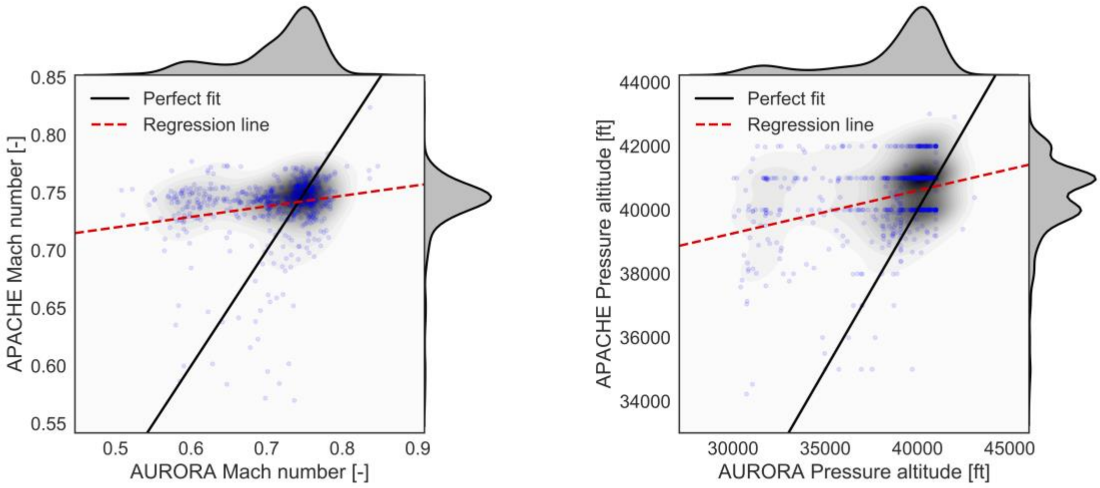
Regarding the cruise Mach and altitude, Fig. F-19 shows that in the regions of more density of flights (around 0.73 for the Mach number and 41000 ft for the altitude) both APACHE and AURORA TPs select very similar speeds and altitudes. In the other regions, the APACHE TP selects higher Mach numbers and altitudes. This Figure agrees with the results presented before, in which the APACHE TP shows, in general, shorter flight times. Figure F-19 also shows that the cruise Mach number of APACHE follows a clear Gaussian distribution, while for AURORA the distribution presents a long tail down to Mach numbers around 0.5. The cruise altitude distribution of AURORA TP is very similar to that of the Mach number. Conversely, for the APACHE TP the cruise altitude distribution shows an irregular distribution, with three peaks where most of the trajectories are concentred.



(a) Scatter plot

(b) Histogram

Figure F-18 Trip time comparison for Test Case 3



(a) Cruise Mach (median)

(b) Cruise altitude (median)

Figure F-19 Cruise Mach and Altitude comparison for Test Case 3

F.2.6 Conclusion

The aim of this document was to present the results of a validation activity, in which the trajectories generated by the APACHE trajectory predictor (TP), developed by the Technical University of Catalonia (UPC) were compared with those obtained by a similar tool used in the AURORA Project, developed by Boeing Research and Technology Europe (BRTE). The validation exercise consisted in three different Test Cases. For each Test Case, the cruise altitude, cruise speed, fuel consumption, flight time and ground distance figures of a set composed by 1500+ trajectories representing realistic traffic were compared.

Results from Test Cases 2 and 3, in which the lateral route was optimised in a hypothetical full free route scenario (from origin to destination), show **that the ground distance of APACHE and AURORA trajectories is very similar, validating in this way the lateral route optimization routines of both APACHE and AURORA TPs.**

Regarding the vertical profile, results from **all Test Cases show that both APACHE and AURORA TPs make use of the same weather data, validating their respective weather data processing and modelling functions**. Yet, some differences in the altitude and speed profiles are observed, which, in turn, lead to discrepancies in the fuel consumption and flight time. Several factors have been identified as potential causes of such differences.

The principal reason of the differences between APACHE and AURORA flights is the fact the **aircraft performance models i**mplemented by these TPs is not exactly the same. On the one hand, the AURORA TP implements the Base of Aircraft Data (BADA) v3.x. On the other hand, the APACHE TP implements BADA v4.2.

Secondly, the **modelling of Continuous Cruise Climb (CCC) operations** is performed differently. The AURORA TP models a CCC as a completely unconstrained flight, from take-off to landing, lacking from speed and altitude restrictions and neglecting typical operational procedures and/or Air Traffic Management (ATM) constraints. The APACHE TP models the CCC as composed by three segments: climb, cruise and descent. The climb segment is performed at maximum climb thrust setting and is split in several phases. The initial phases bring the aircraft to 250 kt of Callibrated Airspeed (CAS), the well-known maximum speed restriction enforced below FL100. Then, the climb continues at constant CAS up to FL100, where an acceleration phase is modelled to achieve the most optimal climb speed. The climb is then performed at constant CAS up to the crossover altitude. From that altitude on, the climb phase is performed at constant Mach until the top of climb. The phases composing the descent segment, which is performed at idle thrust, are exactly the same but in the reversed order. **Therefore, in APACHE, the CCC is only applied in the cruise phase, from the top of climb to the top of descent,** not following any flight levels allocation and orientation scheme and not considering the minimum rate of climb constraint.

Another factor that, to a letter extend, could lead to differences are the **different initial and final conditions of the optimisation problem**. The APACHE TP starts the climb and finishes the descent at operationally realistic speeds (as a function of the aircraft model), while for the AURORA TP the initial and final speeds are free variables of the optimisation problem.

Finally, another contribution to these differences are the variety of logics and optimisation methods employed by both TPs. The AURORA TP computes the vertical profile solving a constrained optimal control problem, while the APACHE TP makes use of look-up tables, pre-computed off-line, to select the optimal speeds.

Besides the observed differences, it can be concluded that both APACHE and AURORA TPs are capable to generate realistic traffic scenarios following similar patterns, both in the vertical and the lateral domain. In addition, the fuel consumption, ground distance and flight time figures required for ATM performance assessment are also analogous.

Apache

APACHE consortium

UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

ALG TRANSPORTATION INFRASTRUCTURE & LOGISTICS

DIVISION OF AIRPORTS AND AIR TRAFFIC SAFETY
FACULTY OF TRANSPORT AND TRAFFIC ENGINEERING
UNIVERSITY OF BELGRADE

ENAC
ECOLE NATIONALE DE L'AVIATION CIVILE