Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

DEGREE FINAL PROJECT

TITLE: Scoring Mechanism for Automated ATC Systems

DEGREE: Bachelor's degree in Aerospace Systems Engineering

AUTHOR: George-Daniel Movilă

ADVISOR: Enric Pastor Llorens

DATE: February 20th, 2023

Title: Scoring Mechanism for Automated ATC Systems

Author: George-Daniel Movilă

Advisor: Enric Pastor Llorens

Date: February 20th, 2023

Abstract

Given the exponential expansion of air transport that pressures the existing infrastructure and makes the current systems over-capacitated, it is no surprise that automated Air Traffic Control (ATC) Systems are on the rise. These systems will improve the situational awareness of Air Traffic Controllers (ATCOs) while reducing their workload and potentially allowing for capacity increases. An example of such a system is the prototype ATC Real Ground-Breaking Operational System (ARGOS), entirely designed by EUROCONTROL Maastricht Upper Area Control Centre (MUAC).

The system is evolving daily, and it will become MUAC's most intelligent agent to support ATCOs in their decision-making process. It is built to take complete care of basic Controller-Pilot Data-Link Communications (CPDLC), i.e., flights logged onto CPDLC not involved in any complex scenario, and to provide support for complex traffic scenarios. The current version provides conflict detection tools and suggests conflict-free trajectories.

The prototype lacks a metrics-based system to evaluate its performance to be able to tune its parameters; thus, the focus of this project is to design a scoring mechanism to assess its performance based on the instructions given to the pilot. The first part is represented by the development of a concept that incorporates the current safety criteria in ATC and introduces new performance criteria relevant to ARGOS (and ATC). In contrast, the second part is a posteriori analysis of the output log files of the prototype that store the coordinates of all current simulated flights along with the commands given by ARGOS.

Two types of scores were considered. The D-Score or the decreasing score measures the efficiency of ARGOS concerning operational safety. In this case, a score of 100% is perfect. It is conceptualized according to safety ATC requirements. The I-Score or the increasing score measures the inefficiency of ARGOS, so higher is less effective. It provides information related to the performance of the trajectory proposed by ARGOS.

I wish to express my profound respect and appreciation for the advisor of this project, Prof. Enric Pastor Llorens, and the ACE/MOSYS teams at Maastricht UAC for their ideas and contributions to this project.

I convey my sincere gratitude to my family and my girlfriend for their ongoing support during these years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

ACC	Area Control Centre
ACT	Aircraft Count
ADEP	Departure airport
ADES	Destination airport
ADS-B	Automatic Dependent Surveillance-Broadcast
AFL	Actual Flight Level
ANG	Convergence Angle
APW	Area Proximity Warning
APW	Airspace Penetration Warning
ARGOS	ATC Real Ground-Breaking Operational Systems
AI	Artificial Intelligence
ASM	Airspace Management
ATC	Air Traffic Control
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
ATS	Air Traffic Services
CAA	Civil Aviation Authority
CAIT	Controlled Airspace Infringement Tool
CFL	Cleared Flight Level
CPA	Closest Points of Approach
CS	Call sign
D-Score	Decreasing Score
DAIW	Danger Area Infringement Warning
DCT	Direct-to Instruction
ECAC	European Civil Aviation Conference
ECL	Enroute Cruising Level
FFL	Filed Flight Level
FL	Flight Level
FRA	Free Route Airspace
HDG	Heading Instruction
HMI	Human-Machine Interface
I-Score	Increasing Score
IAS	Indicated Airspeed
ICAO	International Civil Aviation Organization
ICFL	Intermediate Cleared Flight Level
IFR	Instrument Flight Rules
KPI	Key performance indicator
LFV	Air Navigation Services of Sweden
ML	Machine-Learning
MTCD	Medium-Term Conflict Detection
MUAC	Maastricht Upper Area Control Centre
RAI	Restricted Area Intrusion
RFL	Requested Flight Level
RRP	Route Re-joining Point
RVSM	Reduced Vertical Separation Minimum

SES	Single European Sky
SESAR	Single European Sky ATM Research
SI	Separation Infringement
STCA	Short-Term Conflict Alert
STR	Airspace Structure
TBP	Turn-Back Point
TFL	Transfer Flight Level
TRA	Temporary Reserved Area
TSA	Temporary Segregated Area
UAC	Upper Area Control Centre
VFR	Visual Flight Rules
VRC	Vertical Rate of Change
VSM	Vertical Separation Minimum
XCOP	Exit Coordination Point
XFL	Exit Flight Level

INTRODUCTION

In the context of the continuous expansion of air transport, more and more people travel between countries faster and safer than ever before with increased accessibility, greater comfort, and lower prices. This growth has placed pressure on the air traffic management (ATM) system which ensures the safe and efficient flow of air traffic. As the complexity of air traffic has risen, the result is a system that is becoming increasingly congested, leading to potential delays and inefficiencies. To address these challenges, there is a need for continued development and improvement of the efficiency and capacity of ATM services. Efficient ATM operations are essential for ensuring that airlines can operate with maximum reliability and cost-effectiveness while minimizing the environmental impact of air travel; on the other hand, capacity refers to the maximum number of aircraft that can be safely accommodated within specific airspace or airport over a given time period. Efficiency and capacity are interdependent, meaning that improvements in one can lead to improvements in the other. Efforts to improve efficiency and capacity in ATM are ongoing, with continued research and development focused on improving the performance of the technology and system currently used, as well as adopting more collaborative and data-driven approaches to manage air traffic. Moreover, significant advancements in decision-support systems in ATM have been made in recent years; these systems use advanced algorithms and data analytics to support air traffic controllers in making informed decisions about the flow of air traffic.

Air Traffic Control (ATC) is the main ground-based service of ATM provided by air traffic controllers (ATCOs). ATC is resposible for the safe and efficient flow of air traffic by ensuring compliance with established regulations and procedures; in addition, ATC provides clear and concise communication between pilots and coordinates with other personnel to provide a rapid response in the event of an emergency [1]. The ATC system is under a lot of pressure to adapt and manage the growing air traffic demand safely, effectively, and financially feasible [2]. Even during routine tasks, air traffic controllers are confronted with challenging workloads, and their job carries a lot of responsibility [3]. As a result, it is no surprise that automation in ATC has been increasingly important in recent years as the volume of air traffic has grown. Moreover, just like the ATM system, it is generally agreed that the future of the ATC system will evolve towards higher levels of automation [4]. This project will contribute to the automation of ATC by developing a scoring mechanism for automated ATC systems and will implement it for one of the leading European Air Navigation Service Providers (ANSPs): EUROCONTROL Maastricht Upper Area Control Centre.

Maastricht Upper Area Control Centre (MUAC) is an international air navigation service provider operated by EUROCONTROL on behalf of four States – Belgium, Germany, Luxembourg, and the Netherlands, providing multinational civil and military air navigation services in the upper airspace of the aforementioned countries, being a member of the Functional Airspace Block Europe Central (FABEC). Since the impeccable cooperation between the German Air Force and MUAC in 1975, new concepts and technologies started to be developed and implemented at the agency, most of which were later adopted throughout the continent. From Short-Term Conflict Alert (STCA) in 1980 which gives controllers a 128-second warning of possible infringements of minimum separations standards, to the first in the world implementation of the Controller-Pilot Data-Link Communications (CPDLC) via the Aeronautical Telecommunications Network (ATN) or Europe's first initial 4D (i4D) trajectory management flight, MUAC has become the first cross-border civil-military air navigation service provider in Europe [5].

MUAC's concept of operation for the following years (CONOPS 2030) presents the ambition to provide ATCOs with a high level of automated support, assisting them in data-driven decision-making. Predictive models will be used by the systems to deliver solutions based on historical and real-time data [6]. Their latest software developments include the ATC Real Ground-Breaking Operational System (ARGOS) which is becoming the most intelligent assistant of MUAC's ATCOs. The system provides conflict detection tools and suggests conflict-free trajectories [7].

ARGOS is designed to reduce the workload of ATCOs by taking care of basic CPDLC flights, i.e., flights logged onto CPDLC and not involved in any complex scenario; on the other hand, at complex traffic scenarios, the system provides support for controllers in their decision-making process [8]. Moreover, capacity increases are expected with using ARGOS as an intelligent agent. Thus, ARGOS is fully present in MUAC's Automation Strategy: “Let ATCOs focus on the real, challenging work, to do what they are the best at, and leave the routine work to the machine” [9].

Scoring an automated system like ARGOS is essential because it allows to measure its performance against predetermined criteria, and ensures that it is meeting the expectations and goals of the organization. Evaluating ARGOS's performance would help identify where the system needs improvement and where its strengths are; it would also provide feedback to developers and end-users (ATCOs). Therefore, the goal of this final degree project is to develop a metrics-based system for evaluating the performance of ARGOS. The project aims at identifying certain criteria that can enable the understanding of the system's behavior. Moreover, the scoring aspires to be used as a tuning-mechanism for future system upgrades.

Chapter 1 introduces the theoretical background of ATC as well as methods used to measure performance in ATM. Chapter 2 presents automated systems that are currently under development and introduces the technologies used by ARGOS. Chapter 3 is devoted to the study of a mechanism to evaluate the performance and safety of ARGOS based on output data. Chapter 4 presents the results of the scoring mechanism. Chapter 5 includes the conclusions and further improvements of the project.

CHAPTER 1 ATC PRINCIPLES

1.1 Introduction to Air Traffic Management

The first powered flight was made by the Wright brothers in 1903, and since then, aviation has come a long way. ATM originated in those early days of aviation when pilots had to rely on visual cues to avoid collisions. In 1920, it enabled air transport in unfavourable meteorological conditions. This system worked well enough when there were only a few aircraft in the sky, but as air traffic increased, it became clear that a more formalized system was needed. A more advanced ATM system was developed after World War II, and it relied on ground-based radar to track aircraft. It was used successfully for many years, but it had its limitations. Today, air traffic management is a complex system that relies on advanced technologies such as data communications and advanced surveillance systems.

According to ICAO, ATM is “the dynamic, integrated management of air traffic and airspace including air traffic services, airspace management, and air traffic flow management – safely, economically and efficiently – through the provision of facilities and seamless services in collaboration with all parties and involving airborne and ground-based functions” [10]. ATM incorporates three main services: air traffic services, air traffic flow management, and airspace management.

With Air Traffic Flow Management (ATFM), control sector congestion is avoided by regulating the flow of air traffic as efficiently as possible. It ensures that demand matches supply by imposing restrictions on traffic flows; thus, taken measures are considered pre-tactical [11].

Airspace Management (ASM) optimizes the airspace to ensure the safety of civil and military users. It is the process of organizing, planning, and allocating air routes and sectors.

Air Traffic Services (ATS) are divided into three main services. The air traffic control (ATC) service expedites and maintains the orderly flow of air traffic and provides guidance to pilots through clearances. The flight information service provides advice and information useful for the efficient and safe conduct of flights. The alerting services notifies appropriate organizations in case of emergency [12].

Air traffic control provides area, approach, and aerodrome control services depending on the flight phase. Aerodrome control centers are located in control towers at airports while approach control centers can be found below the airport control tower or at a remote place. Area control centre (ACC), also known as en-route centre, handles the traffic in pre-defined airspace sectors outside the approach control areas. In most cases, airspace boundaries correspond to countries’ territorial boundaries.

1.2 Separation standards

The Convention on International Civil Aviation, also known as the Chicago Convention, was a convention of representatives from 54 countries, who convened to develop a set of global standards and practices for civil aviation. The Chicago Convention resulted in the creation of the International Civil Aviation Organization (ICAO) which sets standards for aircraft separation to ensure the safety and efficient use of airspace. The specific standards belong to Annex 2 – Rules of the Air [49], Annex 11 – Air Traffic Services [12], and Annex 6 – Operation of Aircraft [50].

The standards cover various types of separation, including lateral, longitudinal, and vertical. The type of airspace and aircraft will determine the precise difference needed. Longitudinal separation involves maintaining a certain distance between aircraft on the same flight path, depending on the aircraft type and airspace. In airspace where radar is used, aircraft are separated by 5 nautical miles.

According to the ICAO 4444 document [10], lateral separation can be obtained by demanding that aircraft fly on specified tracks with a minimum separation angle between them. Depending on the type of navigation assistance used, both aircraft must be set on radials or tracks that diverge by a specific amount, and at least one aircraft must be at a minimum of 15 NM away from the facility [20].

Longitudinal separation ensures that the spacing between the aircraft's estimated position is over a certain minimum distance. Methods like speed control and the Mach number technique may be applied to aircraft of the same or diverging tracks to maintain longitudinal separation. For aircraft on the same track, ICAO specifies 15 minutes separation or 10 minutes when frequent speed and position determinations can be made using navigation aids . The same criteria apply for aircraft flying on crossing tracks i.e. intersecting portions or tracks other than same/reciprocal tracks. Reciprocal tracks are opposite/intersecting tracks with an angular difference of more than 135 degrees but less than 225 degrees [10].

When Automatic Dependent Surveillance-Broadcast (ADS-B) or radar is used, the minimum separation is 5 nm. In some exceptional cases, it may be reduced to 3 NM, given the capacities of the ATC monitoring systems [20].

Vertical separation involves maintaining a specific altitude difference between aircraft. According to same ICAO document, the Vertical Separation Minimum (VSM) shall be a nominal 1,000 ft (300m) below FL 290 and a nominal 2,000 ft (600 m) at or above this level. Under a designated airspace subject to a regional agreement, the vertical separation may be lowered from 2,000 ft to 1,000 at levels below FL 410; above this level, the separation remains 2,000 ft. This reduction of the VSM is called the Reduced Vertical Separation Minimum (RVSM). Additionally, only flights that are RVSM-approved are allowed to fly at the reduced standard [21].

If the separation is compromised and aircraft are on a collision course, the Traffic Alert and Collision Avoidance System (TCAS) issues warnings or guidance to pilots. The technology is enabled if only both aircraft are TCAS equipped.

1.3 ATC Clearances and techniques

An ATC clearance is an authorization from an ATCO to an aircraft to proceed by a specified route, altitude, and speed. It is usually issued as a radio transmission from an ATC facility to ensure the efficient flow of aircraft. Clearances are essential, and pilots must adhere to them strictly. They may be modified or revoked at any time, and the pilot must maintain awareness of all changes. Violating an ATC clearance can result in severe consequences, including loss of life. A pilot may not deviate from an ATC clearance except in an emergency or response to a TCAS advisory.

Clearances are issued for the different flight phases, including departure, approach, and en-route. Departure clearances ensure that aircraft take-off in a controlled manner, without interfering with other aircraft or ground operations; they include runway clearances, which specify the runway the aircraft should use for take-off or clearances that allow the aircraft to commence take-off. Approach clearances authorize the aircraft to fly a specific approach procedure to land on a particular runway at a specific airport; they include information such as runway number, approach type, etc. En-route clearances are instructions given to aircraft to navigate through controlled airspace, maintain specific altitudes and speeds, avoid certain areas, and communicate with ATCOs at designated frequencies. The most common types of en-route clearances are:

- Direct clearances – allow the aircraft to fly directly to a specified destination without following a predefined route. This type of clearance is often used to expedite a flight’s arrival at its destination or to avoid congested airspace.
- Altitude clearance – specifies the altitude at which an aircraft must fly. This clearance is critical for maintaining safe separation between aircraft at different altitudes. The clearance may specify a specific altitude, a range of altitudes, or a specific flight level.
- Speed clearance - specifies the velocity at which the aircraft must fly. It is used to maintain separation between aircraft, as well as optimize fuel consumption and reduce delay.
- Holding clearance – is issued when an aircraft cannot land at its destination airport due to congestion, weather conditions, or other factors. The clearance specifies a holding pattern for the aircraft to follow.

In the en-route phase, ATC uses different techniques to manage the flow of air traffic, including vectoring, speed control and flight level changel.

1.3.1 Vectoring

ATC offers aircraft a navigation service called aircraft vectoring. According to [10], vectoring provides navigational guidance to aircraft through specific headings using an ATS surveillance system. The technique is mainly used in the en-route phase to help the aircraft maintain the desired track.

As a general rule and unless otherwise indicated, from the limit of the airspace the controller is responsible for, aircraft cannot be vectored over half of the separation minimum. Some standard methods to vector aircraft are heading locking and sequencing. The first one is occasionally employed when there is adequate space between aircraft, but it is slightly above the required minimum; in this cases, the controller “locks” the headings of the two aircraft, so that they continue on the current heading. Sequencing is usually achieved by combining speed control and vectoring to achieve the desired distance. Moreover, opposite conflicts are also solved through vectoring; separation is usually achieved with relatively small heading changes [17].

1.3.2 Control of Speed

ATCOs use speed control as a key technique to manage the flow of aircraft traffic. By regulating the speed of aircraft, safe operations are ensured while improving the overall efficiency of the airspace. For instance, an aircraft approaching a busy airspace or airport may be instructed to slow down to conform to a specific traffic pattern or to avoid congestion.

ATCOs also use speed control to optimize the flow of traffic. By adjusting the speed, it can be ensured that aircraft are spaced at optimal intervals, reducing delays and increasing capacity in the airspace. For example, by slowing down an aircraft flying too fast, more space can be created for other aircraft to manoeuvre, reducing the risk of congestion and delay.

Another important aspect of speed control is the use of speed restrictions. They can be imposed in specific airspace areas for safety, security, or environmental reasons. For example, they may be set in areas with a high density of aircraft or in areas with sensitive ecological systems to reduce noise pollution.

The ground speed affects an aircraft’s future location (and, subsequently, separation). The Indicated Airspeed (IAS) and Mach number are used to attain the necessary ground speed as it is not possible to use them directly. At lower altitudes (typically below FL 250), aircraft use IAS as the primary reference. At higher altitudes (above FL 250), the speed of sound is lower; because aircraft usually fly at the upper limit of their speed they will eventually have to switch from the IAS limit to remain under the Mach limit. Speed modifications should be specified in multiples of 0.01 Mach at levels at or above FL 250. At levels below FL 250, speed changes should be set in multiples of 10 kt, depending on IAS. Moreover, below 10,000 ft the speed is limited to 250 knots [19].

When using the speed control, the following elements must be considered: aircraft class, wind direction and speed, flight phase, and aircraft level. Even with all these considerations, aircraft may not be able to comply with speed control instructions due to restrictions such as aircraft performance, limitations, weather conditions, or other safety considerations. In such cases, pilots will communicate with ATC to negotiate an appropriate solution.

Speed control is a common task for air traffic controllers, and it typically adds a relatively low workload; it is often the most effective method for sequencing traffic. frequently the most effective method for sequencing traffic.

1.3.3 Level changes

Flight level changes are frequently used by ATCOs to solve conflicts. The rate of climb or descent is measured in feet per minute (ft/min) or meters per second (m/s). The usual rate of climb or descent is 1000ft/min. In some cases, ATC may authorize the pilot’s request for a specific climb or descent profile. This is usually requested when aircraft have performance constraints.

A vertical clearance may include vertical speed clearance, or it may be a separate one. It may also contain the upper or lower limit of the vertical speed, a condition (e.g., before reaching a level or point), or more information (e.g., a reason). Too frequent clearances for changes in climb/descent rates should be avoided [18].

The advantages of choosing a flight level over other techniques are: it requires small interventions as the aircraft keeps flying while using its own navigation and follows the planned route; separation is achieved faster; the climb/descent is easier to monitor in the radar display. The drawback is reduced flight efficiency if the difference between the desired flight and the cleared flight level is high; moreover, the level change may require coordination with upper or lower sectors [51].

Usually, a climb is preferred instead of a descent because it results in better flight efficiency. Despite that, if the aircraft is unable to climb due to weight, or it is approaching its top of descent (ToD) the descent is considered the better option.

1.4 Free Route Airspace Concept

Free route airspace (FRA) is the airspace where users can freely plan their route using defined entry and exit points while remaining under local air traffic control supervision. In collaboration with military and civil specialists in airspace design, and relevant international agencies, EUROCONTROL launched the development and implementation of the FRA concept. Until now, it has significantly improved flight efficiency by reducing fuel consumption and flight time. Moreover, it improves traffic predictability by offering stable trajectories and optimizes how conflict detection tools are used [22].

With FRA, the structure of the fixed airway is eliminated, and airways with a set of predetermined fixes, i.e., entry, arrival, intermediate, exit, or departure, are used instead of airspace blocks. Users can freely arrange a route in this area without consulting the airways network and instead adhere to straightforward flight rules such as using: a departure or an entry fix to enter the FRA; arrival fix or exit to leave the free route; intermediate fixes when following flight plan definition rules or avoiding non-flight zones. Therefore, the main difference is that aircraft are not required to follow traditional fixed routes; FRA provides more flexibility to airlines and controllers and allows them to adapt to changing weather patterns and traffic flow; also, reduced fuel consumption and flight times translate to significant cost savings for airlines [23].

FRA was first implemented in the airspace of the member states of the Functional Airspace Block Europe Central (FABEC), which includes Belgium, France, Germany, Luxembourg, the Netherlands, and Switzerland. FRA has also been implemented in the airspace of the Baltic states. In the airspace of the Danube Functional Airspace Block (FAB), FRA is partially implemented, meaning that is enabled during the night hours. In addition, Other European countries including Austria Croatia, Denmark, Finland, Italy, Poland, Portugal, Spain, Sweden, and the UK have either already implemented FRA in certain parts of their airspace or are planning to do so in the near future.

Nevertheless, the current implementation limits the benefits of the free route. Structural limitations, i.e., national borders, time limitations, or traffic flow restrictions, decrease flight efficiency of the actual free route capability. Solutions proposed include unifying airspace blocks, optimized airspace configurations, and cross-border operations [23].

1.5 Traffic Conflict Detection Systems

According to [24], conflict is a violation of the separation minima of the two-converging aircraft; conflict detection is “*the discovery of a conflict as a result of a conflict search”*; the conflict search is *“computation and comparison of the predicted flight paths of two or more aircraft for the purpose of determining conflicts.”*

It can be argued that one of the most important parts of an ATCO’s job is to detect conflicts. To recognize the prospective incidents, ATC uses a set of ground-based conflict detection systems that are designed to detect potential conflicts between aircraft or to inform the ATCOs about the events that may compromise safety. After properly identified, the controller shall solve the conflict using proper techniques, as the one described in the previous sections. Many factors can influence the detection of a conflict, including recurrent training, discipline, fixed-route/free-route environment, or system support [25].

Thus, developing software capable of handling air traffic with human-like performance requires a proper understanding of conflict detection systems. Moreover, assessing the performance of such software - which is the focus of this thesis – shall prioritize safety. Ground-based conflict detection systems used for the en-route phase by ATC are as follows:

- Short-Term Conflict Alert (STCA) is part of the ground-based safety assistance tools used in ATC to inform the controllers of a potential or actual infringement of the separation minima. It is integrated into the controller working position (CWP) and generates warnings when two or more aircraft are on a collision path. It is intended to generate visual and/or audible alerts in the short term, for up to 2 minutes. It is rather a concept than a unique system, and each ANSP is responsible for establishing a clear STCA policy [26].
- Medium Term Conflict Detection (MTCD) is a flight processing system with multiple functions, including trajectory prediction, conflict detection, trajectory update, and trajectory edition. It can be configured to detect potential conflicts, loss of required separation, and violation of segregated airspace for up to 20 minutes of look-ahead time [28].

- Area Proximity Warning (APW) is another ground-based safety net intended to inform the controller of unauthorized penetration of airspace, by promptly producing an alert of a prospective or actual violation of the necessary spacing to that airspace. Like STCA, it provides a 2-minute look-ahead time before penetrating danger/prohibited/restricted area or controlled airspace. Other names for the same system are Airspace Penetration Warning (APW), Restricted Area Intrusion (RAI), Danger Area Infringement Warning (DAIW), and Controlled Airspace Infringement Tool (CAIT) [27].

1.6 ATCO Workload

The importance of automation in ATC has been demonstrated in recent years as the volume of air traffic has grown. It is expected that the ATC system and ATM will evolve toward higher levels of automation [4]. Without further automation, the workload of a controller will be pushed to unknown limits. They must ensure the safe flow of air traffic, and they carry high responsibilities. Their alertness shall be unquestionable, and the workload must be accurately determined for the best efficiency. Low workload levels imply inefficient application of resources, while high levels overstretch their capacities [29].

The common tasks of ATCOS are handover (flight receiving into the sector, along with the sub-task it implies like sending/receiving confirmations/ flight strips), conflict detection, entry clearance, flight level clearance, horizontal track clearance, speed clearance, conflict resolution, and handoff. Depending on the levels of automation, these tasks may be performed in an optimized way [30].

The workload of a controller is a complex function that has many variables. In [31], NASA developed a complexity measurement tool where individual flights contribute to sets of proper characteristics or airspace sectors. The overall complexity is calculated using 13 factors with individual coefficients. The values of the coefficients are related to the severity of the induced workload. The paper suggests that aircraft density, aircraft proximity near sector boundary, complex conflict geometries, and coordination among sectors generate most of the workload. Therefore, an automated system ought to be capable of reducing the number of aircraft that the ATCOs manage, handling complex conflicts, and coordinating with other sectors.

1.7 ATM Performance

Single European Sky (SES) is an initiative to modernize, re-organize and improve the congested and inefficient European airspace. The project has been going on for more than 20 years. The technological pillar of SES is the Single European Sky ATM Research (SESAR) program. SESAR defines, creates, and implements technology to transform the ATM in Europe [42] [43].

SESAR and SES use performance schemes to measure, monitor, and validate their projects. The SESAR performance ambitions for 2035 for controlled airspace are presented in the European ATM Master Plan; the specific performance areas are capacity, cost efficiency, operational efficiency, environment, safety, and security. They are measured through different indicators, also called key performance indicators (KPIs). Examples of KPIs include departure delay (min/dep), accidents with direct ATM contribution (#/year), ATM-related security incidents resulting in traffic disruptions, etc. One or more KPIs are linked to a specific key performance area. For instance, the gate-to-gate direct ANS cost per flight KPI is connected to the cost efficiency KPA, the goal of which is to reduce the ATM services unit cost by 50% [44].

Other agencies have also implemented methodologies for measuring airspace efficiency. NATS – the British ANSP – and the UK Civil Aviation Authority (CAA) implemented the 3-dimensional inefficiency (3Di) metric used as a KPI to measure the progress of their CO_2 emissions target. 3Di is similar to one of the scores that will be defined in this project (I-Score) as it runs from 0, meaning that efficiency is good, to 100+, translating to lousy efficiency. In the 3Di metric, each flight is scored according to six quantifiers [45].

The criteria used in the 3Di metric are:

- Climb: Continuous climb from the ground to cruise levels give a score of 0. The scores have values between 2 and 10, depending on the time spent in level flight which is counted as inefficient.
- Cruise: A 0 score is given when the flight achieves the last Filed Flight Level (FFL) or higher levels. Scores range from 3 to 9 and are conditioned by the level below FFL that the aircraft cruise.
- Descent: A perfect score is provided when aircraft continuously descend from cruise to ground. Periods of level flight in the descent phase negatively affect the score, which varies from 3 to 11.
- Holding: As holding cause vertical and horizontal inefficiencies, the score fluctuates between 14 and 51 in relation to the time spent holding.
- Horizontal track (UK FIR): The score varies between 1 and 24, and it is affected by the additional track mileage compared to the great circle distance.
- Horizontal track (whole flight): The impact of airspace interfaces on additional track mileage alters the score from 1 to 15 [45].

EUROCONTROL uses the Horizontal Flight Efficiency Indicators to measure the en-route additional distance with respect to the great circle distance. The supplementary distance can be compared with the actual trajectory or the last filed flight plan, which are subject to the Key performance Environment indicator based on the Actual trajectory (KEA) and Key performance Environment indicator based on the last filed flight Plan (KEP), respectively. The KPIs are expressed as a ratio of the length of trajectories and achieved distances [46].

En-route Vertical Flight Inefficiency (VFI) is another KPI proposed by EUROCONTROL. It compares the maximum altitudes in flight plans between specific airport pairs with the maximum altitudes of flights between similar airport pairs. VFI is calculated from a single flight perspective. Factors such as meteorological conditions and aircraft operators' policies may affect the accuracy of the VFI [47].

CHAPTER 2 AUTOMATED ATC SYSTEMS

As highlighted in the previous section, automation in ATC has become a need, and in some very complex and overflown airspace, it is not an option. Full automation in ATC is unlikely to be implemented in the following years. Yet, steps were already taken for this matter as agencies became more concerned about the growing demand for ATS. For this project, the automated ATC systems refer to systems capable of handling air traffic not involved in complex traffic scenarios. They are not developed with the purpose of replacing ATCOs, but rather to enhance their productivity and reduce their workload by taking care of basic air traffic.

2.1 Next-generation ATC systems

The current ATC systems heavily rely on human controllers and have limitations in terms of capacity and efficiency. However, with the technology advancement, the next generation of ATC systems is being developed, which will overcome these limitations and improve the overall ATM. Some key features encompass data integration for a comprehensive and real-time view of the airspace, collaborative decision-making, performance-based navigation for increased navigation accuracy, and digital communications to reduce information exchange errors. Depending on modernization, these components can be mapped to different levels of automation.

The Levels of Automation Taxonomy offer a concise view of a long-term evolution towards increasing automation levels and perhaps full automation [33]. Fig. 1 depicts the level of automation in ATC in four cognitive stages. For instance, the actual system support at MUAC for the executive and coordinating controllers are represented by the yellow square, while its size is the number of tasks and their frequency. MUAC’s automation objectives for basic traffic are highlighted in blue circles; the objectives for complex traffic scenarios are shown in green circles.

Automated systems provide high automation levels to support ATC. The goal of these systems is to reduce the workload on human controllers, improve accuracy, increase efficiency, and enhance safety. By automating certain functions such as aircraft separation and traffic sequencing, these systems can help reduce the risk of human error and increase the capacity of the airspace. However, the level of automation in these systems must be carefully balanced with the need for human oversight and intervention to ensure safety in all situations. Projects such as Advanced Autoplanner at the Swedish ANSP, Skyler from AIRTC company, or ARGOS developed by MUAC contribute to the next generation of ATC systems.

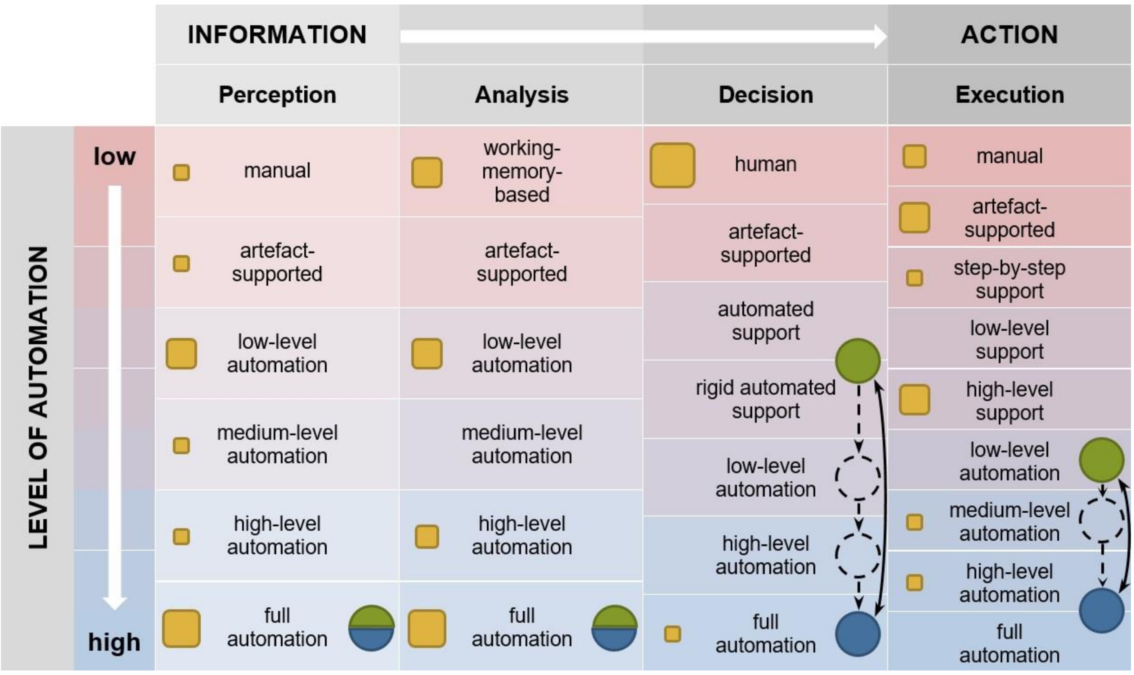


Fig. 1 Levels of automation in ATC in four different stages [9]

2.1.1 Advanced Autoplanner (AAP)

Advanced Autoplanner is a project initiated by the Air Navigation Services of Sweden (LFV) and IBM to prove the concept of fully automated ATC systems [34]. They developed an Artificial Intelligence (AI) model for the en-route flight phase to provide ATC clearances. The AI Model determines the locations of the aircraft in real time and analyses the future state of the airspace. It uses a specific AI method (lattice-based search space exploration technique) for determining safe actions. Then the identified safe actions are ranked according to preferences (e.g., increasing altitude as it improves fuel efficiency). Simulations showed that the model is capable of handling traffic in the Swedish airspace at normal and increased traffic density with *"only a small number of occurred conflicts"* due to data lacking. Future improvements mention the machine-learning (ML) models to re-rank actions for fine-grained aircraft characteristics. LFV also plans to explore tactical planning capabilities, restricted areas, and the weather component [35].

2.1.2 Skyler

Skyler is *"an Artificially Intelligent Air Traffic Controller agent"* which gives instructions to flights; thus, it performs the tasks of an ATCO. The agent was developed by AIRTC, a company involved in the automation and augmentation of ATC. It is meant to pave the future of automated ATC for the Upper Airspace. The AI can work in automated ATC mode, where it takes full control of the traffic, and in augmented ATC mode, where it becomes a partner of the controller; the human controller can choose which flight Skyler may control. As with all automated ATC agents, its core relies on data-link so that instructions can be uploaded into the pilot's cockpit. Interestingly, AIRTC claims that the same output will be generated with the exact same traffic situation, though it uses AI. Moreover, they used MUAC's Hannover UIR airspace as a development and test sector. Impressive results were achieved in controlled simulations with over 200k flights, with an average occupancy of 55 aircraft and zero safety issues [36].

2.1.3 ARGOS

ARGOS stands for ATC Real Ground-Breaking Operational System and it a system designed to provide ATCOs with a high level of automated support; it incorporates conflict detection tools and suggests conflict-free trajectories. Its purpose is to allow for capacity increases by facilitating the work of the controller. Entirely designed at MUAC, ARGOS is meant to take complete care of flights that are not involved in complex traffic scenarios by sending clearances via CPDLC. In complex traffic scenarios, it helps the controllers by offering tools and advisory in the decision-making process. The approach method is deterministic compared to the stochastic methods used for the previous systems.

There are two ARGOS versions ongoing at MUAC. The production ARGOS will be implemented within the human-machine interface (HMI) and controller-working position (CWP); its level of automation is very low. The second ARGOS version is a prototype independent of the HMI and CWP; it has a very high level of automation, and it is used to test and design the new algorithms for the production version.

2.2 ARGOS Assistance tools

ARGOS program implements current conflict detection tools and introduces new assistance tools to facilitate the work of the controller. By combining the tools provided by ARGOS, several services can be deployed to operational users. The scoring mechanism provides insight into these solutions, therefore scoring metrics can be used to fine-tune the following tools:

2.2.1 Probe tool

As presented in [32], the Probe function used in various ATC systems is implemented in the HMI along with the other clearance tools (level allocation, trajectory modification) and allows the controller to test vertical clearances without changing the system. After selecting the Probe function in the interface, a new trajectory is generated and sent to the MTCD subsystem (section 1.5). MTCD analyses the produced trajectory and informs the controller, as shown in Fig. 2.

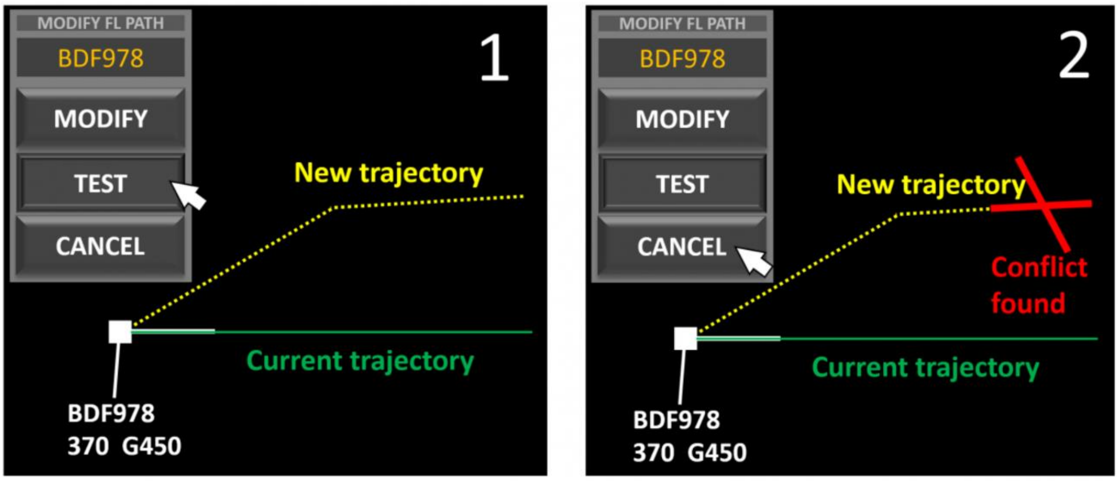


Fig. 2 a. Selction of a trajectory in Probe tool; **b.** Potential conflict found with Probe tool [32]

Along with the presented functionalities, the Probe function designed for ARGOS provides automatic conflict detection without intervention from the controller. Moreover, it provides a mouse-over functionality without selecting the flight levels but with a reduced look-ahead time of 2 minutes. It is provided by the MUAC's NTCA (Near-Term Conflict Alert) - a system similar to MTCD. The conflict display highlights the conflicting target flights.

2.2.2 LORD (Lateral Obstacle and Resolution Display)

Because the Probe function provides only vertical conflict detection, a new tool was developed to probe the headings – LORD. The on-request conflict detection of the Probe tool is triggered in the same way LORD is, while the conflict display acts as a circle around the track direction rather than highlighting the conflicting flights. The track position is the center of the circle/arc, and the probed conflicting headings are highlighted with different colours depending on the time before the conflict.

2.2.3 ARGOS Solutions

ARGOS Solutions contain the most sophisticated algorithms in the ARGOS program due to their trajectory calculations. While the Probe and LORD tool provide tactical conflict detection and assistance for the controllers in their critical decision-making process, ARGOS Solutions can be seen as pre-tactical. These are comprehensive solutions with conflict-free trajectories for a finite look-ahead period. They fulfill the following conditions:

- Join the flight plan at a predetermined point;
- Adhere to trajectory validity rules (conflicts, sector boundaries, restricted areas)

ARGOS Solutions are crucial for the system; therefore, the validity of the trajectories will be analysed in the scoring.

In the horizontal plane, two types of trajectories are generated by analysing the flight plan:

- Planning horizontal trajectory
- Tactical horizontal trajectory

2.2.3.1 **Planning horizontal trajectory**

The purpose of the planning horizontal trajectory is twofold. Firstly, if possible conflicts are detected, it either vectors the flight through a heading instruction, or provides a Direct-to (DCT) instruction. Secondly, to-rejoin the flight plan route at a specified point, it provides a future heading or DCT instruction

The planning horizontal trajectory contains two parts. The first part starts at the track position and ends at a so-called Route Re-joining Point (RRP), i.e., a fixed-point on the flight plan route. The second part is the remainder of the flight plan route starting at the RRP, as shown in Fig. 3 [8].

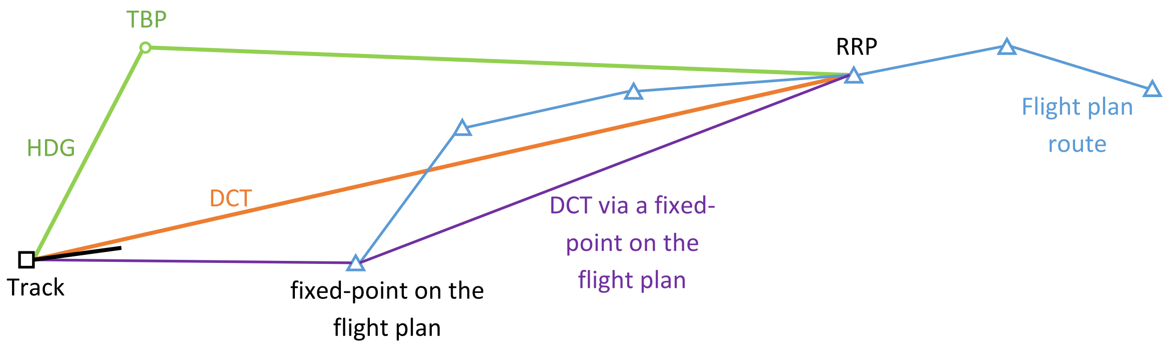


Fig. 3 Planning horizontal trajectory examples

The first part depends on the presence of a direct (DCT) or heading (HDG) instruction:

- If a DCT instruction is given, it can be of the following types:
  - Direct to RRP;
  - Direct to RRP via a fixed-point as an intermediary point on the flight plan route;
- Otherwise, it is a two-leg trajectory defined as follows:
  - The first leg is parallel to the instructed heading
  - The first leg joins the track position and a so-called Turn-Back Point (TBP);
  - The second leg joins the TBP and the RRP

2.2.3.2 **Tactical horizontal trajectory**

In cases where only HDG instructions are given, the planning horizontal trajectory incorporates a Turn-Back Point (TBP), which serves as a point at which the aircraft is planned to be turned back to the RRP. However, in the event of a communication failure between the pilot and ARGOS system, the aircraft will continue on its current heading instead of making the planned turn at the TBP. To mitigate the risk of such situations, a tactical horizontal trajectory is computed in addition to the planning horizontal trajectory. This tactical horizontal trajectory consists of a straight line (parallel to the first leg of the planning horizontal trajectory) and does not have a defined endpoints, as depicted in Fig. 4 [8].

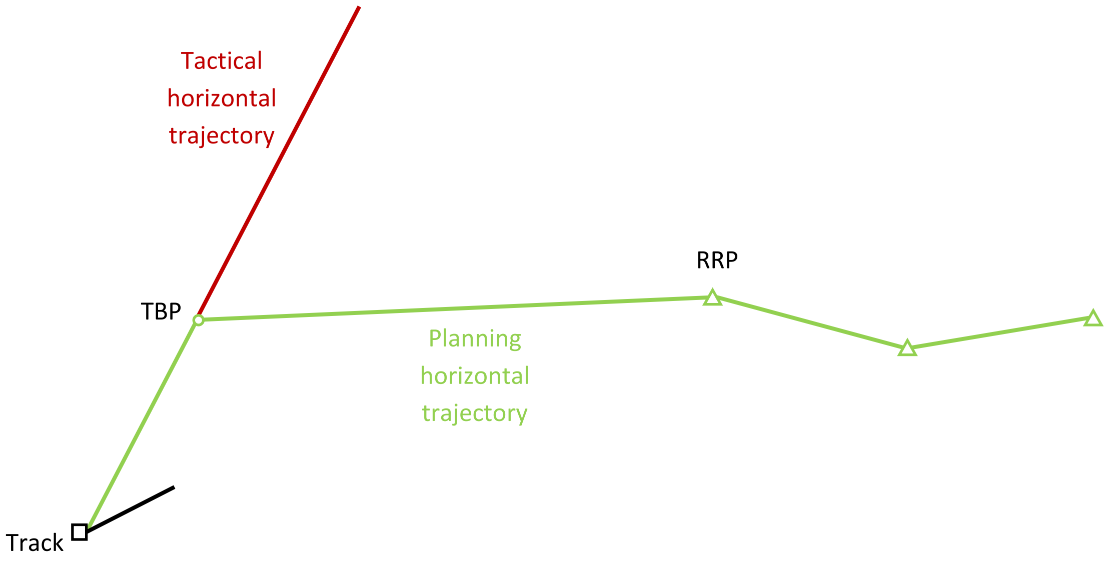


Fig. 4 Tactical horizontal trajectory (in red) to supplement the planing horizontal trajectory (in green) in case of CPDLC loss

The planning/tactical horizontal trajectory is valid if:

- It is conflict-free for a limited look-ahead time (3 minutes);
- It does not intersect sector boundaries;
- It does not penetrate any restricted areas.

CHAPTER 3 SCORING MECHANISM

An automated ATC system uses computer algorithms to manage and direct aircraft in the airspace. The goal of such a system is to improve safety, efficiency and capacity; thus, a scoring mechanism would be used to evaluate its performance and safety. There are several factors to consider when designing a scoring mechanism for automated ATC systems, including the efficiency of aircraft routing, number of collisions or near collisions avoided, compliance with ATC regulations, reduction in flight delays, or success rate in maintaining safe separation between aircraft. The mechanism can be based on a combination of these factors and can be used to make adjustments to the system's algorithm and improve its performance. Additionally, the scoring mechanism should be adaptable, able to update, and over time be accurate and reliable.

3.1 Importance of the posteriori analysis

Scoring automated ATC software is important because it allows us to measure the system's effectiveness and identify areas for improvement. The rating can be done in a number of ways, including through simulations, real-world testing, and monitoring system functionality during live operations.

The importance of scoring the system according to an A posteriori analysis of its output files is that it allows system evaluation based on actual data. It enables the identification of patterns and trends in the system's performance. The information can be used to identify areas for improvement and to develop strategies for further optimizations.

In some cases, A posteriori analysis can also be useful for identifying and troubleshooting system errors, identifying areas where the system may be underperforming and evaluating the system's compliance with regulatory standards.

It also enables the understanding of the system's behavior over time and the identification of any change in the system's quality that may occur. This can help determine issues caused by system upgrades, changes in the operating environment, or other factors.

Overall, a scoring mechanism for automated systems is a critical component that plays an important role in ensuring the safety, efficiency, and reliability of the system. By monitoring and reporting on the system's performance in a variety of areas, the scoring can provide valuable insights that can be used to improve the system's performance over time. As the technology for autonomous ATC systems improves, so should the scoring algorithm.

3.2 Scoring data

The scoring is subject to the prototype version of ARGOS, which simulates the ATCOs main screen in the browser using pre-defined flight plans. After running a simulation, it produces two separate log files. The first log file contains data about the instructions given to each flight, as seen in Table 30 from Appendix A. The most important data is the time when the clearance was given, the planned entry and exit points, and the clearance type. Clearances may be of six types:

- Assume (ASM) – The instruction is sent when the flight is intercepted in the airspace meaning that from now on, the aircraft navigates under ARGOS' control.
- Cleared Flight Level (CFL) – The flight is instructed to climb/descend to the mentioned flight level using vertical speed adjustments.
- Exit flight-level (XFL)[1] – It is a CFL with the value of the planned exit level.
- Heading change (HDG) – The flight is instructed to turn in a specified direction.
- Direct-to (DCT) – The flight is instructed to resume its navigation to the exit point. It is usually given after a HDG or CFL.
- Exit assume (XASM) – The instruction is given when the flight arrives at its exit point (or near) and it is no longer under the system's control.

The second log contains the coordinates for each flight every 4.8 seconds (it is the time required for radar to update aircraft position), as shown in Appendix A, Table 31. It also gives information about the aircraft's velocity. From now on, the instructions log file will be called the scoring file/log; the one containing coordinates will be named the update file/log.

3.3 Scores

The scoring represents an A posteriori analysis of the mentioned data; therefore, the system's performance will be evaluated by using several metrics. Each metric will contain an individual score that will be added/subtracted to generate a final score. Individual scores for metrics provide a more granular evaluation of performance and help identify specific areas where improvement is needed. Moreover, individual scores can make the scoring process more objective, as it relies on less subjective opinions or judgments and more on specific data points and provide a high level of accountability, as they allow for a more precise evaluation of individual criteria contributions.

In ATC, safety is of paramount importance. Performance is also essential, but a scoring mechanism shall be able to differentiate between safety and performance issues. Hence, the proposed method seeks two types of scores: the first score measures safety and it decreases as safety issues are detected, while the second score is related to performance, and increases as deviations from the optimum trajectory are determined. By including a score that decreases as safety is compromised, the scoring ensures that safety issues are immediately flagged and addressed. With the two scores, the scoring system can ensure that each

---

[1] XFL is used to denote both the instruction (cleared to XFL) and the value of the exit flight-level.

safety or performance issue is addressed appropriately and that efforts to improve performance do not come at the expense of safety. The two scores can encourage improvement in both areas, driving overall system optimization and providing a more comprehensive evaluation of system performance, allowing for a more nuanced understanding of the strengths and weaknesses of the system and identifying areas for improvement. The two scores are described as follows:

- Decreasing score – named D-Score – measures operational safety and adherence to separation standards. In this case, a score of 100 translates to a safe trajectory. Besides the minimum ATC requirements, the D-Score analyses critical instructions provided by ARGOS Solutions (tactical trajectories). A concise example of a metric that would affect this score is the number of separation infringements (SI). The principle of the D-Score is that each simulation starts with a score of 100, and as the simulation progresses, score penalties are applied.
- Increasing score – named I-Score – measures the inefficiency of ARGOS; a higher score translates to a less efficient trajectory. It provides performance-related information that can be used for parameter tuning. The distance flown would be a good example of a metric that can affect this score.

3.4 Score penalties

The scores may be affected by two penalties:

- Global penalties. A single serious event (such as separation infringement) triggers a global penalty from the score, which is permanent. They affect only the D-Score.
- Per flight penalties. They are applied to each flight in particular; thus, the score will depend on the number of specific events detected relative to the number of flights, i.e., a simulation with 100 flights, of which 10 generate penalties incurred per flight, will have a better score than a simulation with 100 flights with 50 penalties incurred per flight. They affect only the I-Score.

Each criterion/metric will affect the scores according to their penalties. Considering that the D-Score starts at 100 and I-Score at 0, the final scores are computed with formulas:

$$D-Score = \max\left(100 - \sum_{i}^{n}\sum_{j}^{MD} global\_penalty_{ij}, 0\right) \tag{3.1}$$

$$I-Score = \sum_{i}^{n}\sum_{j}^{MI} \frac{per\_flight\_penalty_{ij}}{nr\_flights\_analyzed_{j}} \tag{3.2}$$

- n = total number of flights
- MD = number of metrics in D-Score
- MI = number of metrics in I-Score
- $global\_penalty_{ij}$ = global penalty of metric **j** for flight ***i***

- $flight\_penalty_{ij}$ = flight penalty of metric **j** for flight **i**
- $nr\_flights\_analyzed_{j}$ = total number of flights that can be analyzed for metric **j**

Instead of computing the weighted sum with the total number of flights, the $nr\_flights\_analyzed_{metric}$ variable was introduced to obtain a deeper insight into the performance. Any ARGOS simulations starts with 0 flights, and as the simulation progresses, flights become visible on radar as they approach the airspace. The simulation can end at any time, thus, some flights might not have completed their trajectories, i.e., were not cleared to the exit point. Thus, situations such as determining the ratio between the minimum distance and the actual flight distance or deviation from the exit point must be normalized according to the number of flights that can be analysed.

3.5 Metrics

Metrics provide a quantitative measure of the performance of a system or algorithm, allowing for comparison between different systems or different versions of the same system. Table 1 shows the suggested metrics and their grouping according to the scores. The following sections will describe each metric in part. The proposed metrics may be used to score any automated ATC systems; their parameters and penalties are intended only for the scoring of ARGOS.

Table 1 Scoring metrics

Quantifier	Description	Score type
AIRPROX	Violation of separation standards (separation infringement)	D
TSA penetration	Penetration of a Temporary Segregated Area	D
STCA	Trigger of a short-term conflict alarm	D
Critical horizontal clearance	Not respecting horizontal safety buffers after HDG	D
Critical vertical clearance	Not respecting vertical safety buffers after reaching CFL	D
Critical ICFL	Not respecting vertical safety buffers prior reaching CFL	D
Track leaving AoR	Flights outside the controlled airspace	D
Mismatched exits	Flights that exit through other points than specified in flight plan	D
Inefficient clearances	Too frequent clearances given to a flight	I
Exaggerated turns	Turns too high used to solve conflicts	I
Inefficient path	Flown more than 5% above the shortest distance	I

Inefficient climb to ECL	More than 3% of the flown distance prior to Top of Descend (ToD) not cleared to the cruising level (ECL)	I
Horizontal deviation	Horizontal deviation from the exit point	I
Vertical deviation	Vertical deviation from the exit point	I
Mismatched clearances	Instructions given to other points than the ones specified in the flight plan	I
Inefficient cruise	Flights that do not reach the planned cruising level	I
Inefficient exit	Flights arriving too early at the transfer flight level	I

3.5.1 AIRPROX

According to ICAO, AIRPROX is *“the code word used in air traffic incident reports to designate aircraft proximity.”* ICAO established several classifications for AIRPROX occurrences that have been examined by the proper agencies, depending on the risk of collision [38]. The classification may be applied to actual aircraft proximity in airspace fully controlled by ATCOs or some automated systems. In the case of an existing risk of collision, the event shall be drastically penalized. Otherwise, if the aircraft were not on converging tracks and there was no risk of collision, the event may be ignored.

ARGOS is built to avoid any separation infringements (SI) at any time, even if there was no risk of collision. Thus, any AIRPROX detection indicates a system malfunction; no AIRPROX shall exist, whatever the type of incident.

No flight penalties are applied for this criterion; any AIRPROX detection, regardless of its severity, is unacceptable for ARGOS; thus, the D-Score will be 0.

For ARGOS, which operates above FL 245, ICAO standards shall be used (see section 1.2):

- minimum lateral separation: 5 nm
- vertical separation RVSM: 1,000 ft above FL290 and below FL410
- vertical separation: VSM: 2,000 ft in below FL290 or above FL410

Table 2 shows the necessary penalties for a D-Score equal to 0.

Table 2 D-Score contribution: AIRPROX

Quantifier	**Global penalty**
AIRPROX	100

3.5.2 TSA penetration

A Temporary Segregated Area (TSA) is an airspace allocated and designated for the exclusive usage of a particular user for a predetermined amount of time. Contrary to a Temporary Reserved Area (TRA), where traffic may be allowed to transit under ATC clearance, in TSA, no other traffic is allowed [39].

An aircraft that penetrates the restricted area for more than 2 minutes is considered unacceptable, generating a D-Score of 0. Despite that, if the aircraft penetrates the area for a very short time, it may not necessarily be unsafe. Therefore, the time spent in the TSA will determine the penalty, as shown in Table 3.

Table 3 D-Score contribution: TSA penetration

Quantifier	Global penalty
TSA penetration	$\min\left(100, Time_{spent\_in\_TSA}(s)\right)$

3.5.3 STCA

STCA is the ground-based safety net that gives controllers alerts prior to a potential conflict in less than 2 minutes. When it comes to ATCOs, short-term alerts are a matter of safety. Regarding automated systems, a triggered STCA translates to poor trajectory planning. An active STCA provides insights about short-term collisions, thus, the event will not be penalized as much as an AIRPROX, as shown in Table 4. Moreover, multiple STCAs can be triggered for the same flights in a short timestamp, i.e., the STCA is activated, then inactive, and then again activated. It shall be considered a single event if the idle time of the alarm is less than 10 seconds.

Table 4 D-Score contribution: STCA

Quantifier	Global penalty
STCA	5

3.5.4 Critical horizontal clearance

The critical horizontal clearance is a metric that analyses the validity of the tactical horizontal trajectory, which was presented in section 2.2.3.2.

Fig. 5 shows the flown trajectory of an aircraft in purple. As can be seen, the flight was instructed with two headings (orange points with ‘HDG’ text adjacent)

- The first HDG was given to avoid conflicts;
- The second HDG was given to re-join the flight on its flight plan. This second HDG corresponds to the TBP, where the flight is supposed to be turned back. Therefore, the tactical horizontal trajectory is the blue trajectory which has to be validated.

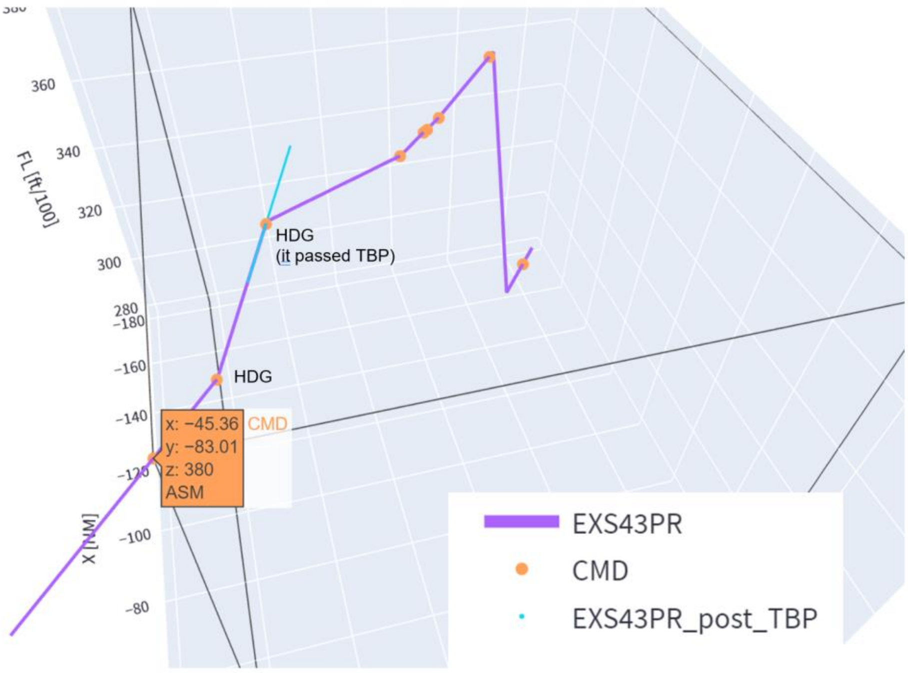


Fig. 5 Example of tactical horizontal trajectory (in blue) to supplement the actual trajectory (in purple)

As defined previously, the tactical horizontal trajectory is valid if:

- It does not generate AIRPROX;
- It does not penetrate TSA;
- It does not cross sector boundaries;

If the tactical horizontal trajectory does not satisfy the mentioned validity requirements, the following penalties are applied:

Table 5 D-Score contribution: Critical horizontal clearance

Quantifier	Scenario	Global penalty
Critical horizontal clearance	AIRPROX	10
	TSA penetration	10
	Sector boundary crossing	5

3.5.5 Critical vertical clearance & Intermediate CFL

Parameters such as aircraft weight and type can influence the rates of climb/descent that aircraft may have. Therefore, when ARGOS issues a level clearance (CFL), it creates an altitude block that must be reserved and protected. The altitude block is created for climbing and descending phases and it is reserved for only one flight. An example of an altitude block is shown in Fig. 6, where the AFL is the Actual Flight Level, i.e., the level at which the aircraft will start climbing or descending, and the CFL is the Cleared Flight Level, i.e., the target flight level. The maximum rates of climb (ROC) or descent (ROD) are shown by the orange lines, and the blue lines show the minimum ROC/ROD.

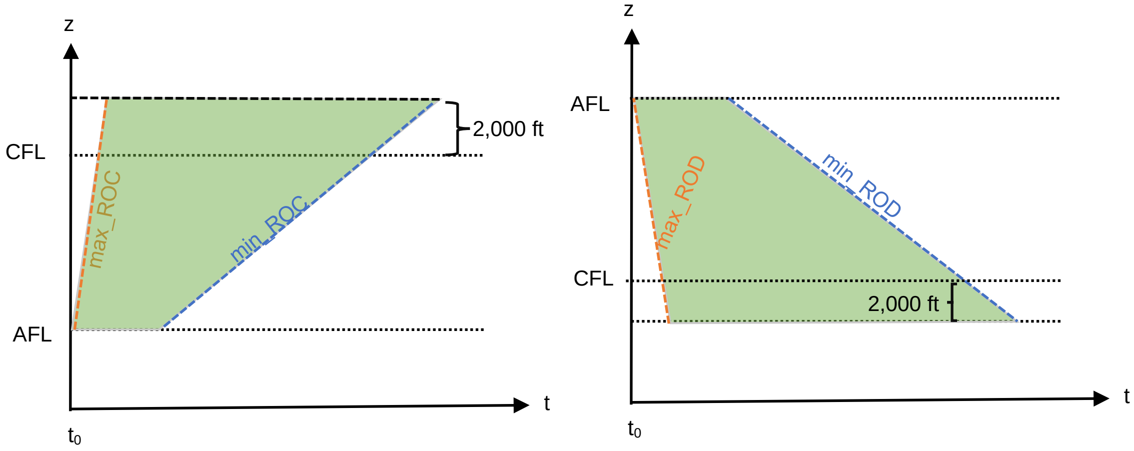


Fig. 6 Altitude block (in green) created for climb/descent:

a. CFL > AFL; **b.** AFL > CFL;

The altitude block is composed of two parts:

- The first part starts at AFL and ends at CFL. It represents the reserved space for aircraft when climbing or descending.
- The second part starts at CFL and ends at CFL + 2,000 ft. It is a safety extension of the first part.

The size of the altitude block is influenced by the possible rates of climb/descent that the aircraft may have. The simulation environment (where ARGOS is tested) sets the maximum rate of climb/descent for all aircraft. Therefore, to determine the actual size and position of the altitude block, the scoring will simulate possible rates of climb/descent, other than the default maximum.

An altitude block is considered valid (protected) if:

- It does not generate any separation infringements/AIRPROX;
- It does not cross any sector boundaries;
- It does not penetrate a TSA.

To determine the validity of the altitude block, two separate metrics will be used, called critical Intermediate CFL (ICFL) and critical vertical clearance:

- The critical ICFL metric analyses the validity of the first part of the altitude block, i.e., the reserved space for the aircraft.
- The critical vertical clearance metric analyses the validity of the second part of the altitude block, i.e., the safety extension.

By using two metrics, it will be easier to identify where safety improvements are necessary. Despite that, both metrics will add the same penalty in the D-Score, because the first part of the altitude block is as important as the other.

The penalties are similar to the critical horizontal clearance, and they are presented in Table 6.

Table 6 D-Score contribution: Critical vertical clearance

Quantifier	Scenario	Global penalty
Critical ICFL	AIRPROX	10
	TSA penetration	10
	Sector boundary crossing	5
Critical vertical clearance	AIRPROX	10
	TSA penetration	10
	Sector boundary crossing	5

3.5.6 Track leaving AoR

The Area of Responsibility (AoR) is the specific geographic region for which the ACC unit is responsible. In general, the AoR may include airports, airspace, and routes, and it is defined by sector boundaries. For ARGOS, the AoR is MUAC's airspace. Leaving the AoR without authorization can have serious consequences. It is completely a matter of safety. Thus, a permanent penalization is applied if a flight is found outside the AoR and it was not cleared to exit.

The penalties are presented in Table 7.

Table 7 D-Score contribution: Track leaving AoR

Quantifier	Global penalty
Track leaving: AoR	20

3.5.7 Track leaving: boundary

Flights that are inside the AoR but are at the boundary or very close to the boundary shall be penalized. They do affect critical safety; thus, the penalization is small compared to leaving completely the AoR, as showin in Table 8.

Table 8 D-Score contribution: Track leaving: boundary

Quantifier	Global penalty
Track leaving: boundary	1

3.5.8 Mismatched exits

Flights that do not reach their planned exit points may pose a risk to other aircraft in the vecinity and to the safe operation of airspace. With automated systems that rely on data-link communications, mismatched exits shall not be allowed. Thus, the penalty is equivalent to the case of AoR leaving. The penalties are shown Table 9.

Table 9 D-Score contribution: Missmatched exits

Quantifier	Global penalty
Mismatched exits	20

3.5.9 Inefficient clearances

As stated in [18], frequent clearances should be avoided as much as possible; they may become inefficient or increase the pilot's workload. Frequent clearances may generate unnecessary delays, which is an important factor in assessing the performance of ATCOs [40]. Thus, an automated system that generates fewer instructions is preferred.

Inefficient clearances affect only the I-Score, and they are desired to be minimized. Only flight penalties are applied. Too frequent clearances are defined as follows:[2]

- 2 horizontal clearances (HDG) given in less than 3 minutes apart.
- 2 CFL given in less than 3 minutes apart
- More than 5 clearances given per 15 minutes flying time.[3]

Table 10 shows the penalties for frequent clearances according to the scenario.

Table 10 I-Score contribution: Inefficient clearances

Quantifier	Scenario	Flight penalty	Flights analysed
Inefficient clearances	2 HDG in less than 3 min apart	50	Flights having at least 1 HDG
	2 CFL in less than 3 min apart	50	Flights having at least 1 CFL
	More than 5 clearances/15 min flying time	60	Flights having at least 1 HDG or 1 CFL

3.5.10 Inefficient path

Efficent trajectories in the en-route phase reduce delays and fuel burned. The possibility of calculating the shortest distances by an automated system is a clear advantage in ATC. Safety is not a concern when scoring the shortest path; only performance is considered. The system should be capable of calculating the shortest distances for each flight, even when conflicting situations appear. Thus, global penalties are not considered in this case; only flight penalties are used.

---

[2] Clearances are calculated per flight and not per simulation.

[3] It considers horizontal and vertical clearances i.e., HDG and CFL. It does not consider ICFL.

Solving conflicts requires a deviation from the initial route; therefore, only flights that have flown more than 5% above the shortest distance are considered for penalties.

Flights that have flown more than 10% of the shortest distance should be penalized more than flights that have flown more than 50% of the shortest distance. Therefore, the penalization is defined as the exceeded percentage of the minimum distance, i.e., if the minimum distance was exceeded by 20%, the penalty is 15, as shown in Table 11.

Table 11 I-Score contribution: Inefficient path

Quantifier	Flight penalty	Flights analyzed
Inefficient path	Percentage of exceeded distance (%)	Flights with complete trajectory

3.5.11 Inefficient climb to ECL

Keeping the aircraft as high as possible is considered good practice in ATC. In MUAC's airspace, flights usually enter the AoR at lower altitudes, climb to the En-Route Cruising Level (ECL), and exit at a lower altitude. As shown in several studies, fuel efficiency increases with altitude [41]. At higher altitudes, jet engines are not required to produce as much thrust because the air is *"thinner,"* which also means less drag. Thus, keeping the aircraft at higher altitudes for as long as possible is a question of performance.

A penalty is applied if the flown distance until the flight was cleared to ECL is higher than 3% of the flown distance until the Top of Descent (ToD). Below that 3% threshold, no penalty is applied. The threshold takes into account the different rates of climb that aircraft may have and also deviations from the flight plan for conflict management. The penalties should be similar to the inefficient path; thus, the flight penalty is the threshold percentage, as shown in Table 12.

Table 12 I-Score contribution: Inefficient climb to ECL

Quantifier	Flight penalty	Flights analyzed
Inefficient climb to ECL	Percentage of exceeded distance (%)	Flights that reached ToD

3.5.12 Horizontal deviation

Deviation from the flight plan is acceptable when solving conflicts. Still, deviation from the exit point in the flight plan is not adequate. Thus, flights having a deviation from the planned exit point higher than 2 nm should be penalized. The penalty is proportional to the deviation it had, as shown in Table 13.

Table 13 I-Score contribution: Horizontal deviation

Quantifier	Flight penalty	Flights analyzed
Horizontal deviation	Value of the horizontal deviation (nm)	Flights with complete trajectory

3.5.13 Vertical deviation

Similar to the horizontal deviation, the vertical deviation shall not be higher than 200ft, the equivalent of 2 flight levels. The penalties are comparable to the previous and are shown in Table 14.

Table 14 I-Score contribution: Vertical deviation

Quantifier	Flight penalty	Flights analyzed
Vertical deviation	Value of the vertical deviation (ft)/50	Flights with complete trajectory

3.5.14 Mismatched clearances

Clearances given specifically to other points than the one in the flight plan shall be penalized. After managing conflicts, the system shall be capable of directing the aircraft to its planned trajectory. The probability of the system to instruct flights with DCT to other points is close to 0, thus, the penalty is higher so that at least one flight can impact the I-Score. Nevertheless, only flight penalties are applied, as shown in Table 15.

Table 15 I-Score contribution: Mismatched clearances

Quantifier	Flight penalty	Flights analyzed
Mismatched clearances	100	Flights with complete trajectory

3.5.15 Inefficient exit

This metric is similar to the inefficient climb to ECL. ARGOS is supposed to maintain the aircraft at higher altitudes if possible (ECL altitude) until it starts descending to the exit point. Thus, the inefficient exit is defined as the exit where the aircraft has reached the Transfer Flight Level (TFL) too early. Current simulations have shown that some flights spent even less than 10 seconds at TFL until they reach the exit coordination point (XCOP). Therefore, a penalty is applied only for flights arriving earlier with more than 30 seconds at TFL, as shown in Table 16. The penalty is proportional to the time (in seconds) spent at TFL. Moreover, only flights that have TFL lower than ECL may be analysed. Flights with ECL equal to TFL or ECL lower than TFL are not subject to inefficient exits.

Table 16 I-Score contribution: Inefficient exit

Quantifier	Flight penalty	Flights analyzed
Inefficient exit	Value of time spent at TFL (seconds)/10	Flights with complete trajectory and Flights that have ECL > TFL

3.5.16 Exaggerated turns

Turns can be classified into shallow, medium or steep. A shallow, or a level turn is characterised by a bank angle of less than 20 degrees, which translate to imperceptible increases in G-load. Medium turns, with bank angle between 20 and 40 degrees still have small G-loads. Steep turns, with a bank angle higher than 45 degrees have noticeable G-loads [48]. Thus, turns bigger than 20 degrees shall be penalized. Moreover, multiple turns higher than 15 degrees in less than 3 minutes are still considered high turns, and they should be avoided. The penalties are proportional with the degrees of the turns, and are presented in Table 17.

Table 17 I-Score contribution: Exaggerated turns

Quantifier	Scenario	Flight penalty	Flights analyzed
Exaggerated turns	Turn higher than 20°	Value of the turn (°)	Flights that have at least 1 turn
	More than 2 turns higher than 15°	Value of the max. turn (°)	

3.5.17 Inefficient cruise

In some cases, the system may instruct the aircraft to fly at lower altitudes than ECL. As stated previously, higher altitudes are preferable in the upper airspace. Thus, flights that didn’t reach the ECL in the flight plan are penalized with the difference between their actual ECL and the planned ECL, as shown in Table 18.

Table 18 I-Score contribution: Inefficient cruise

Quantifier	Flight penalty	Flights analyzed
Inefficient cruise	Difference between actual ECL and planned ECL (FL)	Flights with complete trajectory that have TFL < ECL

3.6 Penalty justification

Penalties based on ratios and percentages were chosen to provide insight into the deviation from the optimum scenario. This category also includes metrics that have values measured in specific units, i.e., degrees, seconds, feet, and nautical miles. For most cases, the penalties were unmodified because as the value of the penalty increases, so does the score. For instance, the higher the degree of the turn, the higher the score would be. The exception for this category is the flight penalty in the vertical deviation, which was divided by 50. The initial division was planned at 100, which was motivated by the fact that 1 FL is equivalent to 100 ft. Still, it created a high discrepancy between the scores results at vertical and horizontal deviation. When it comes to inefficient exit, the value of the time spent at TFL is divided by 10 because around 50% of the analysed flights usually have deviations higher than 30 seconds, which would generate a much higher score for this metric.

The global penalty values are strictly designated to suggest the importance of an event. For instance, a flight found at the boundary is only penalized with a single point because it is not a serious incident (it is not outside the AoR). On the other hand, a critical clearance affects safety but not as much as a violation of separation standards. Thus, if the D-Score is 50, it can be said without looking at the metrics, that the prototype operates at 50% safety.

CHAPTER 4 RESULTS

The objective of the analysis is to compare the score results of the current ARGOS version in different simulation environments, i.e., different speeds at which ARGOS works. In addition, the scores of a previous ARGOS version were compared with the scores of the current version to determine whether the trajectories of the current version are more efficient than the trajectories of the previous software version

The simulation environment runs in real time, but it sets the speed (rate) at which ARGOS processes information. The rates of the simulation environment dictate the printing period of the coordinates/instructions in the log files. Four possible rates may be selected, as shown in Fig. 7.

Fig. 7 ARGOS simulation speed rate selection

The rates are described as follows:

- X1 Rate – It is the default rate at which ARGOS processes information every 4.8 seconds, i.e., the default time at which radar updates the positions of each aircraft. Therefore the coordinates of the aircraft and the instructions given are updated every 4.8 seconds in the logs.
- X2 Rate – At this rate, ARGOS processes information two times faster than at the X1 rate, therefore the printing period of the log data will be double, i.e., every 9.6 seconds.
- X5 Rate – The log data is printed five times faster than the default, i.e., every 24 seconds.
- X10 Rate – The log data is printed every 48 seconds.

The current version was analysed at different processing rates because it allows to evaluate how the software performs under different conditions, specifically how it processes information and generates log files in response to different speeds. By scoring the system at different processing rates, it is possible to observe how ARGOS behaves under various loads and assess its performance. For instance, the prototype may perform well at a lower processing rate but may become unstable or fail at higher rates. Identifying these issues early in the development process can help improve the trajectory performance and stability and prevent potential problems from occurring when it is deployed.

The scoring was conducted by using ARGOS logs provided by MUAC. The set of logs corresponds to the current ARGOS version at the aforementioned rates and a previous ARGOS version at the X1 rate; every log was created using the same flight plans. The statistics derived from the logs at different rates are presented in Table 19.

Table 19 ARGOS logs statistics at different rates

Statistics	**X1 rate**	**X2 rate**	**X5 rate**	**X10 rate**
Total flights	151	151	151	151
Flights with complete trajectory	151	150	151	151
Simulation time	05:25:31	05:25:31	05:25:40	05:20:00
Nr. CFL instructions	92	106	97	105
Nr. DCT instructions	159	159	164	171
Nr. HDG instructions	5	3	4	5
Average turn (°)	18	6.7	12.5	8
Maximum turn (°)	50	10	25	15
Minimum turn (°)	5	5	5	5
Nr. DCT combined with CFL	14	17	15	17
Nr. DCT combined with HDG	0	0	0	0
Nr. CFL combined with HDG	3	2	3	4
Predominant first instruction	DCT	DCT	DCT	DCT
Nr. instructions given to avoid conflicts	40	43	43	45

As can be seen, the logs counted 151 flights over 5 hours and 25 minutes. Surprisingly there were more level changes and directs instructions, and fewer headings given at higher rates when compared to X1 rate. The statistics show that at X1 rate, ARGOS performed worse than at the other rates, as the average turn was the highest. Moreover, there is at least one steep turn at X1 rate, which is a drawback. At X2 rate, the average turn is 6.7° and the maximum turn is 10°, therefore the metric that analyses the turns will have a score of 0 for this rate. Nevertheless, the number of instructions given to avoid conflicts is similar among the rates; the number is between 40 and 45, and it indicates that at least 15% of the total instructions were given to avoid possible conflicts.

The following sections will present the scores obtained for each set of logs at different rates. The last section is dedicated to the comparison between a previous ARGOS version and the current version, both at X1 rate.

4.1 Current version at X1 and X2 Rates

Table 20 shows the number of occurrences of D-Score events and their global penalties for the current ARGOS version at X1 and X2 rates.

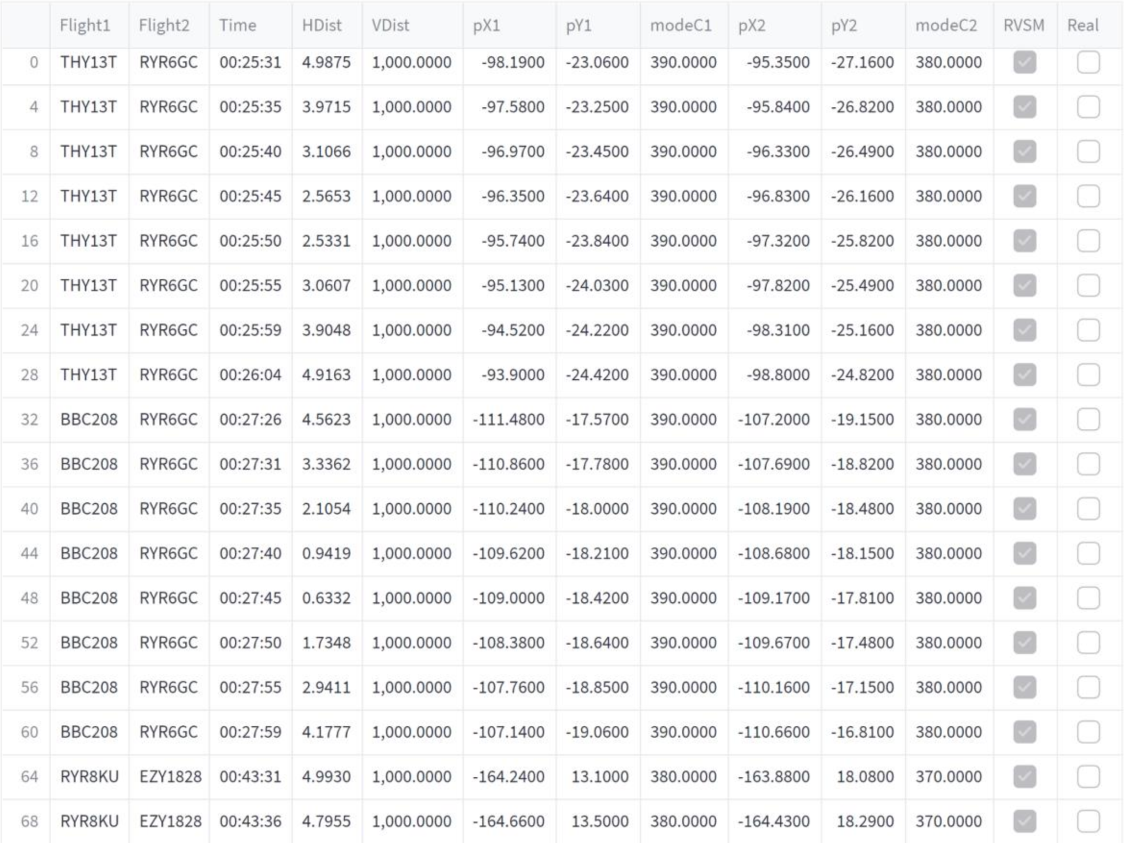
Table 20 D-Score results for the current ARGOS version at X1 & X2 rates

Quantifier	**Occurrences (X1)**	**Global penalty (X1)**	**Occurrences (X2)**	**Global penalty (X2)**
AIRPROX	0	0	0	0
TSA penetration	0	0	0	0
STCA	0	0	0	0
Critical vertical clearance	0	0	0	0
Critical horizontal clearance	0	0	0	0
Critical ICFL	0	0	0	0
Track leaving AoR	0	0	0	0
Mismatched exits	0	0	0	0
Track leaving boundary	6	6	6	6
D-Score	94		94	

For both rates, the D-Score is 94, which suggests that no major incidents were detected. Having a D-Score close to 100 translates to a safe trajectory. Almost all scores were 0, the only parameter affecting the scores is the track leaving boundary metric, with 6 flights close to the sector boundary.

Moreover, no separation infringements were detected. Table 21 shows a small part of the AIRPROX data collected for X1 rate. The data contains sets of two flights that were at the limit, i.e., the horizontal distance between them was below the separation minimum and the vertical distance was exactly RVSM.

Table 21 AIRPROX data for X1 rate



	Flight1	Flight2	Time	HDist	VDist	pX1	pY1	modeC1	pX2	pY2	modeC2	RVSM	Real
0	THY13T	RYR6GC	00:25:31	4.9875	1,000.0000	-98.1900	-23.0600	390.0000	-95.3500	-27.1600	380.0000	☑	☐
4	THY13T	RYR6GC	00:25:35	3.9715	1,000.0000	-97.5800	-23.2500	390.0000	-95.8400	-26.8200	380.0000	☑	☐
8	THY13T	RYR6GC	00:25:40	3.1066	1,000.0000	-96.9700	-23.4500	390.0000	-96.3300	-26.4900	380.0000	☑	☐
12	THY13T	RYR6GC	00:25:45	2.5653	1,000.0000	-96.3500	-23.6400	390.0000	-96.8300	-26.1600	380.0000	☑	☐
16	THY13T	RYR6GC	00:25:50	2.5331	1,000.0000	-95.7400	-23.8400	390.0000	-97.3200	-25.8200	380.0000	☑	☐
20	THY13T	RYR6GC	00:25:55	3.0607	1,000.0000	-95.1300	-24.0300	390.0000	-97.8200	-25.4900	380.0000	☑	☐
24	THY13T	RYR6GC	00:25:59	3.9048	1,000.0000	-94.5200	-24.2200	390.0000	-98.3100	-25.1600	380.0000	☑	☐
28	THY13T	RYR6GC	00:26:04	4.9163	1,000.0000	-93.9000	-24.4200	390.0000	-98.8000	-24.8200	380.0000	☑	☐
32	BBC208	RYR6GC	00:27:26	4.5623	1,000.0000	-111.4800	-17.5700	390.0000	-107.2000	-19.1500	380.0000	☑	☐
36	BBC208	RYR6GC	00:27:31	3.3362	1,000.0000	-110.8600	-17.7800	390.0000	-107.6900	-18.8200	380.0000	☑	☐
40	BBC208	RYR6GC	00:27:35	2.1054	1,000.0000	-110.2400	-18.0000	390.0000	-108.1900	-18.4800	380.0000	☑	☐
44	BBC208	RYR6GC	00:27:40	0.9419	1,000.0000	-109.6200	-18.2100	390.0000	-108.6800	-18.1500	380.0000	☑	☐
48	BBC208	RYR6GC	00:27:45	0.6332	1,000.0000	-109.0000	-18.4200	390.0000	-109.1700	-17.8100	380.0000	☑	☐
52	BBC208	RYR6GC	00:27:50	1.7348	1,000.0000	-108.3800	-18.6400	390.0000	-109.6700	-17.4800	380.0000	☑	☐
56	BBC208	RYR6GC	00:27:55	2.9411	1,000.0000	-107.7600	-18.8500	390.0000	-110.1600	-17.1500	380.0000	☑	☐
60	BBC208	RYR6GC	00:27:59	4.1777	1,000.0000	-107.1400	-19.0600	390.0000	-110.6600	-16.8100	380.0000	☑	☐
64	RYR8KU	EZY1828	00:43:31	4.9930	1,000.0000	-164.2400	13.1000	380.0000	-163.8800	18.0800	370.0000	☑	☐
68	RYR8KU	EZY1828	00:43:36	4.7955	1,000.0000	-164.6600	13.5000	380.0000	-164.4300	18.2900	370.0000	☑	☐

An example of a false positive AIRPROX is exposed in Fig. 8, using the data from the first row of Table 21. The horizontal distance at time 00:25:31 between THY13T and RYR6GC is 4.98 nm, which is below the horizontal minimum; the vertical distance is 1,000 ft, which is precisely the RVSM value. If the vertical distance is below 1,000ft, it would be considered an AIRPROX. It may also be considered an AIRPROX if the flights would have flown under FL290, where the RVSM is no longer applied, but as the flights are above that level, RVSM is considered the separation standard.

No TBPs were associated with headings; thus, there weren't any critical horizontal clearances detected. Yet, there were 92 possible critical vertical clearances, and none resulted in AIRPROX.

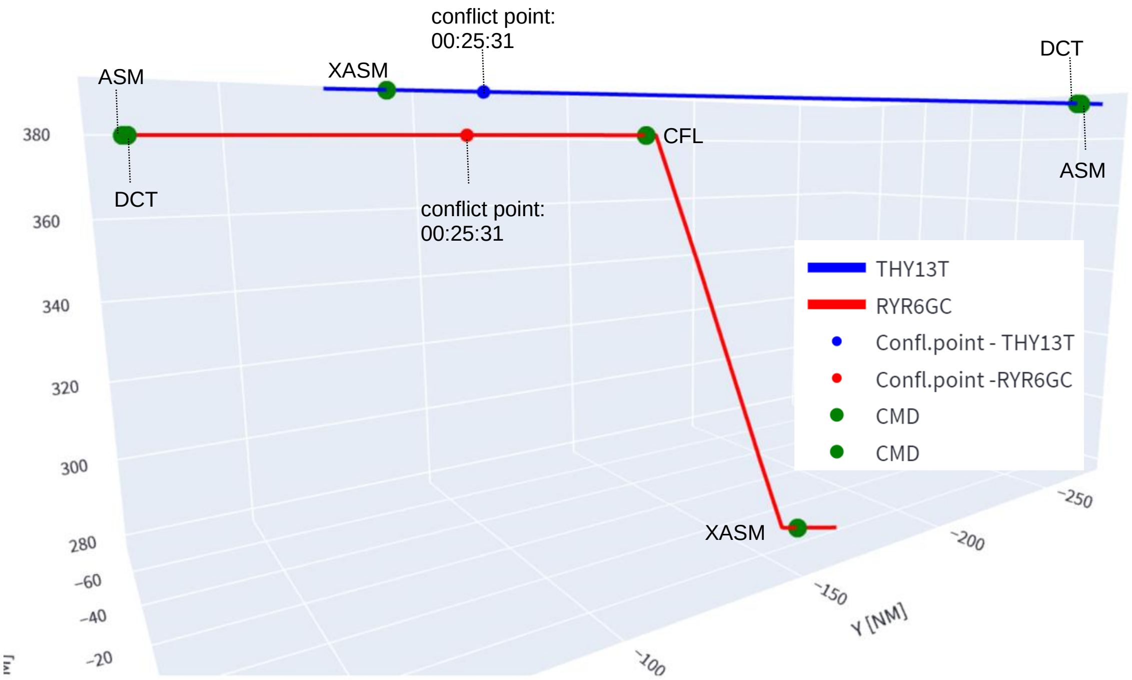


Fig. 8 Trajectory: THY13T, RYR6GC in a false-positive AIRPROX (X1 rate)

On the other hand, the calculated I-Score is 69.7 at X1 rate and 62.7 at X2 rate. The values of each metric are presented in Table 22, where the contributing fights represent the number of flights that directly impacted the I-Score. In contrast, the total number of flights analysed is the number of aircraft that met the quantifier's criteria to be introduced in the score, e.g., when scoring exaggerated turns, only 5 flights could be analysed at X1 rate (because only them had a HDG instruction).

Table 22 I-Score results for X1 and X2 rate

Quantifier	Flights analyzed (X1)	Contributing flights (X1)	Total penalty (X1)	Flights analyzed (X2)	Contributing flights (X2)	Total penalty (X2)
Inefficient clearances	12	1	8.33	13	2	19.23
Inefficient path	151	111	17.58	150	110	17.76
Inefficient climb to ECL	16	3	1.58	17	2	1.41
Horizontal deviation	151	149	13.77	150	150	13.56
Vertical deviation	151	21	6.33	150	19	5.57
Mismatched clearances	151	0	0	151	0	0
Inefficient exit	47	20	4.26	47	19	2.74
Exaggerated Turns	5	2	15	3	0	0
Inefficient cruise	47	2	2.61	47	1	2.45
I-Score	69.7			62.7		

The score for inefficient clearances at X1 rate is 3.33, with only 1 out of 12 aircraft having too frequent clearances. The trajectory of that aircraft is shown in Fig. 9. Flight EXS43PR was assumed into the airspace at time 00:31:16, and after a short time, it received a DCT in combination with XFL instruction. Instead of a continuous descent, ARGOS opted for a conventional descent, probably to avoid conflicts. After five consecutive CFL instructions, it has received yet another one to climb. Finally, the flight was instructed at 00:52:14 to level at its exit flight level. Non-continuous descents increase fuel consumption and reduce performance; thus, a penalty of 50 for a flight with frequent clearances motivates the penalty value.

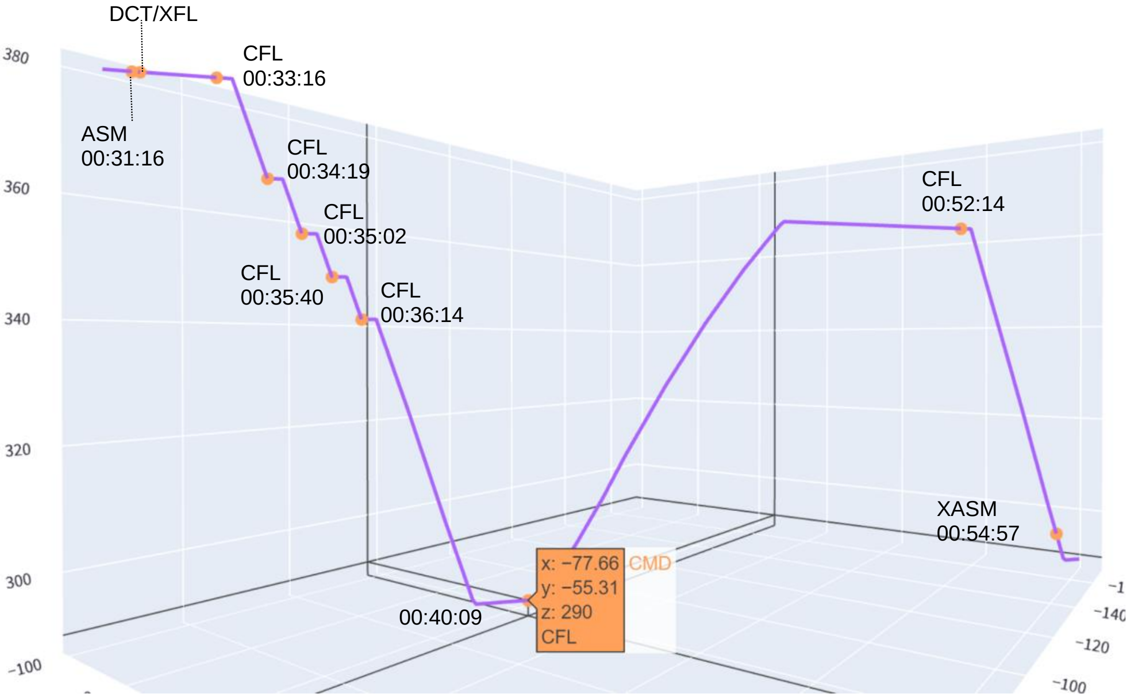


Fig. 9 Trajectory of EXS43PR having too frequent clearances (X1 rate)

On the other hand, the inefficient clearance metric affected drastically the I-Score for X2 rate, as it is more than double when compared to the X1 rate: it had two flights with too frequent clearances. The trajectory of the flight that affected the most this metric is shown in Fig. 10. Flight EXS21CV received 17 CFL instructions. The first 7 CFLs were probably given because the flight didn't adhere to the first instruction, while the others were sent to avoid conflicts. Either the case would be, the frequency of clearances altered the I-Score.

The inefficient path metric contributes the most to the I-Score for both rates. Moreover, 40 flights didn't exceed the 5% threshold. One flight worth mentioning is EWG3590 (Fig. 11) which exceeded the horizontal distance by 153%; its actual distance was 29.8 nm, compared to the minimum distance of just 11.75 nm. Therefore, it is important to set a threshold to certain metrics to limit unacceptable behaviours, such as exceeding the minimum distance by more than 150%.

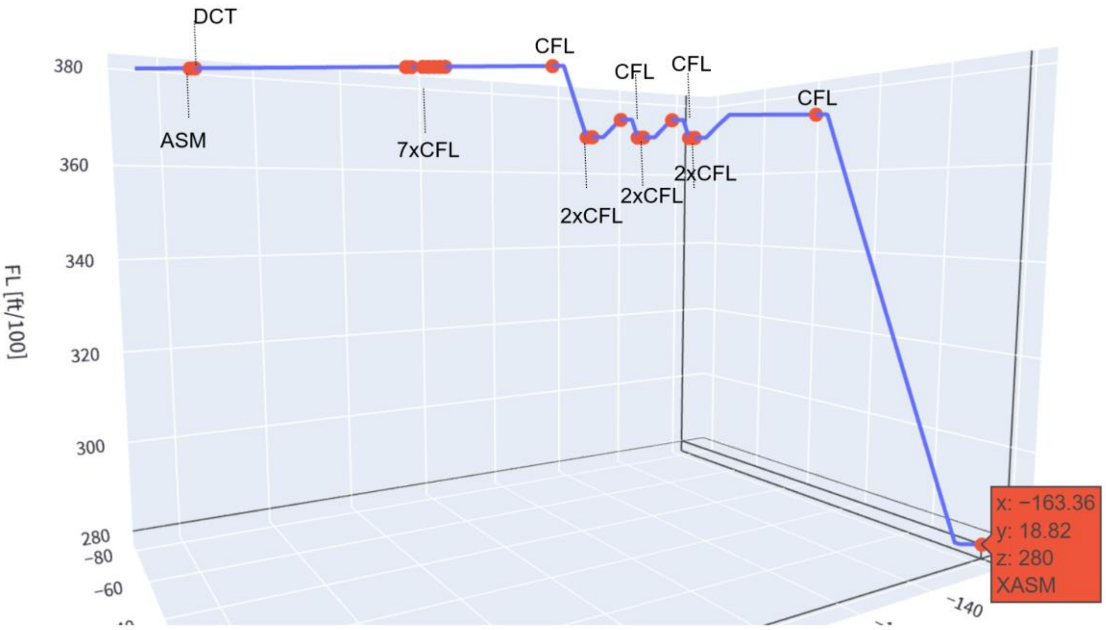


Fig. 10 Trajectory EXS21CV with 17 CFLs (X2 rate)

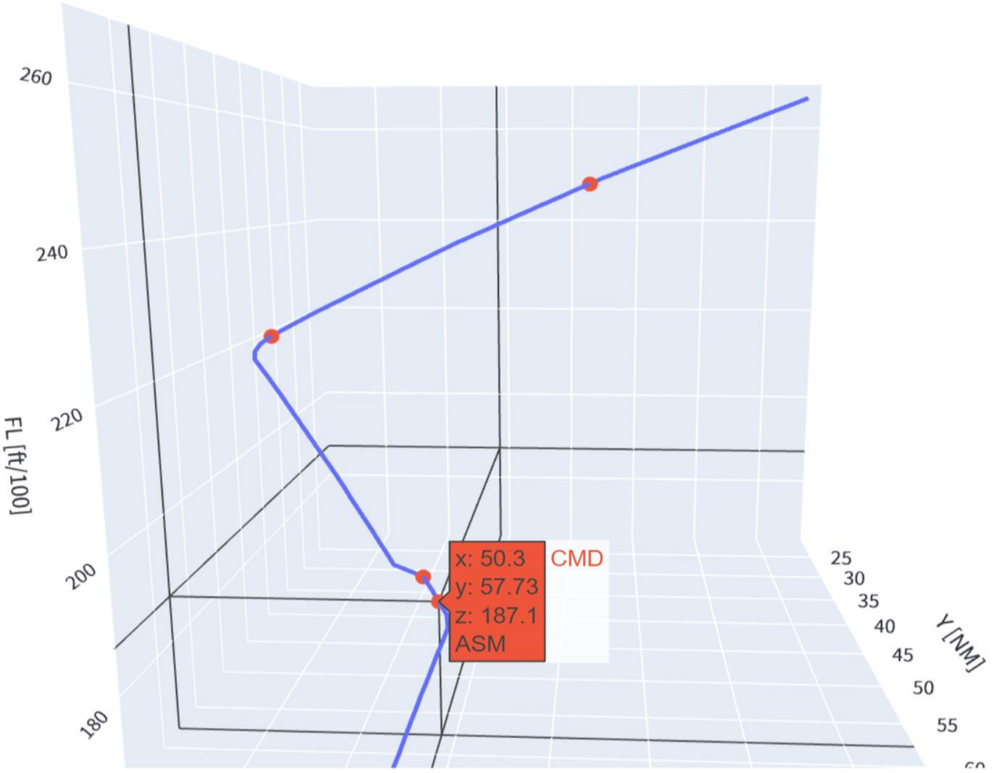


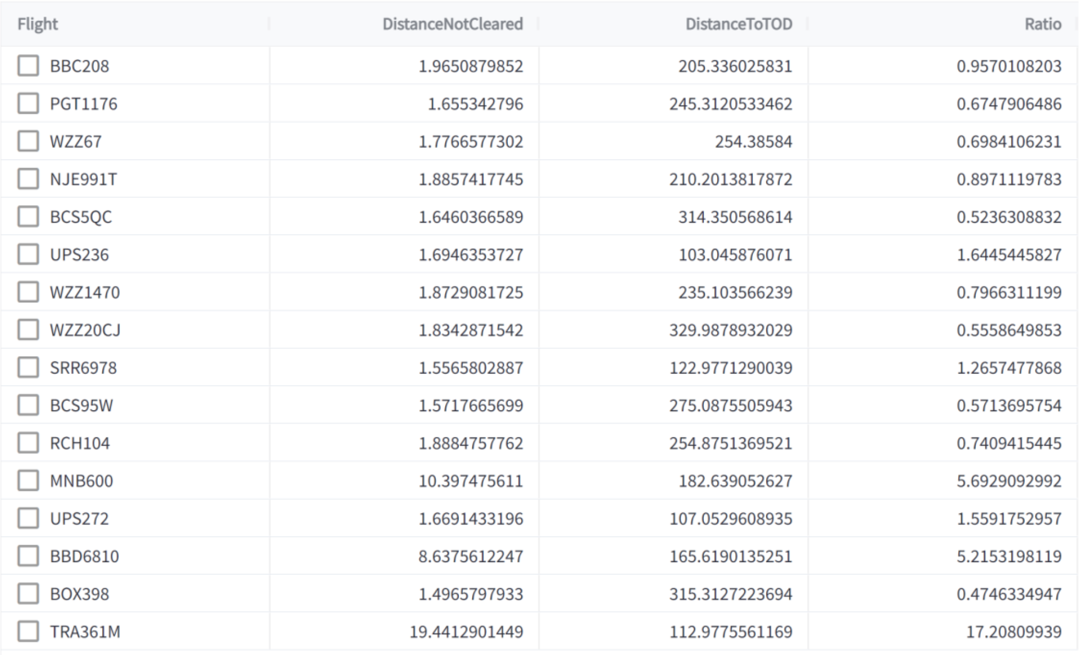
Fig. 11 Trajectory: EWG3590 exceeds the shortest distance by 153% (X1 rate)

Horizontal deviation has impressive magnitudes for both rates: almost all aircraft horizontally deviated from the planned exit point with at least 2 nm. The mean deviation value is 13 nm. Values for vertical deviation are better: only 21 flights at X1 rate and 19 flights at X2 rate had a vertical deviation of more than 200ft.

At X1 rate, two aircraft were instructed with headings higher than 20 degrees (one with 25 and the other with 50), and two aircraft didn't fly at the planned cruising level, while at X2 rate the score for exaggerated turns is 0 and only one aircraft didn't reach the ECL. Nevertheless, there were no clearances given to different exit points.

Surprisingly, the inefficient climb to ECL has the lowest value of the metrics for both rates. It signals that only 1.58% and 1.41% of the distance until ToD was not cleared, at X1 rate and X2 rate respectively, which can be considered a positive point for ARGOS. The distances and the ratios are shown in Table 23 (only at X1 rate). Most flights have a ratio below 1%, and there is only an outlier with 17%.

Table 23 Inefficient climb to ECL data (X1 rate)



Flight	DistanceNotCleared	DistanceToTOD	Ratio
☐ BBC208	1.9650879852	205.336025831	0.9570108203
☐ PGT1176	1.655342796	245.3120533462	0.6747906486
☐ WZZ67	1.7766577302	254.38584	0.6984106231
☐ NJE991T	1.8857417745	210.2013817872	0.8971119783
☐ BCS5QC	1.6460366589	314.350568614	0.5236308832
☐ UPS236	1.6946353727	103.045876071	1.6445445827
☐ WZZ1470	1.8729081725	235.103566239	0.7966311199
☐ WZZ20CJ	1.8342871542	329.9878932029	0.5558649853
☐ SRR6978	1.5565802887	122.9771290039	1.2657477868
☐ BCS95W	1.5717665699	275.0875505943	0.5713695754
☐ RCH104	1.8884757762	254.8751369521	0.7409415445
☐ MNB600	10.397475611	182.639052627	5.6929092992
☐ UPS272	1.6691433196	107.0529608935	1.5591752957
☐ BBD6810	8.6375612247	165.6190135251	5.2153198119
☐ BOX398	1.4965797933	315.3127223694	0.4746334947
☐ TRA361M	19.4412901449	112.9775561169	17.20809939

The current I-Scores show that ARGOS is working well in terms of performance and that the significant difference between the two rates is made by the inefficient clearance metric. Despite that, improvements are necessary for what concerns trajectory planning. Results have shown that many aircraft deviated horizontally from the exit point, with considerable values. Nonetheless, software improvements will be easier to implement using this I-Score as a reference.

4.2 X5 Rate and X10 Rate

D-Score is 93 for ARGOS running at X5 rate and 88 at X10 rate; the results are shown in Table 24.

Table 24 D-Score results at X5 and X10 rate

Quantifier	**Occurrences (X5)**	**Global penalty (X5)**	**Occurrences (X5)**	**Global penalty (X5)**
AIRPROX	0	0	0	0
TSA penetration	0	0	0	0
STCA	0	0	1	5
Critical vertical clearance	0	0	0	0
Critical horizontal clearance	0	0	0	0
Critical ICFL	0	0	0	0
Track leaving AoR	0	0	0	0
Mismatched exits	0	0	0	0
Track leaving boundary	7	7	7	7
D-Score	93		88	

The difference between the scores was made by one STCA alarm, which had a negative impact on the D-Score at X10 rate. The other criteria have a score of 0, except the track leaving boundary metric which had seven occurrences.

It is worth noting that AIRPROX data from Table 25, at X10 rate, shows actual infringement occurrences that were not reflected in the D-Score. They were not counted as SI because the events happened right after the aircraft exited the airspace, as seen in Fig. 12. Both aircraft were below FL290. For MUAC airspace, they do not pose a risk, but it may constitute a potential hazard for the ACC below MUAC.

Table 25 AIRPROX data at X10 rate

	Flight1	Flight2	Time	HDist	VDist	pX1	pY1	modeC1	pX2	pY2	modeC2	RVSM	Real
0	THY13T	RYR6GC	00:26:24	4.9242	1,000.0000	-99.4100	-22.6500	390.0000	-96.3400	-26.5000	380.0000	☑	☐
4	BBC208	RYR6GC	00:28:48	1.8037	1,000.0000	-110.2800	-18.0600	390.0000	-111.1500	-16.4800	380.0000	☑	☐
8	EZY2456	UPS236	00:49:36	3.3622	1,000.0000	-170.6700	23.7500	280.0000	-167.3600	24.3400	270.0000	☐	☑
12	EZY2456	UPS236	00:50:24	2.3090	1,000.0000	-175.5700	27.0500	280.0000	-173.4700	26.0900	270.0000	☐	☑
16	EZY2456	UPS236	00:51:12	2.7893	1,000.0000	-180.6000	30.4400	280.0000	-179.5900	27.8400	270.0000	☐	☑
20	THY4	TOM22B	00:52:00	3.2596	1,000.0000	-118.0600	-16.8500	370.0000	-114.9300	-15.9400	380.0000	☑	☐
28	RYR8KU	EZY1828	00:52:48	4.6929	1,000.0000	-228.3200	33.9700	380.0000	-223.8600	35.4300	370.0000	☑	☐
32	RYR8KU	EZY1828	00:53:36	3.9278	1,000.0000	-233.4400	36.6600	380.0000	-229.7000	35.4600	370.0000	☑	☐
36	EXS43PR	EXS21CV	00:58:24	2.7066	1,000.0000	-174.7500	26.5100	290.0000	-176.8800	28.1800	280.0000	☑	☐
40	EXS43PR	EXS21CV	00:59:12	2.5770	1,000.0000	-179.2400	30.3000	290.0000	-181.7700	30.7900	280.0000	☑	☐
44	MLD832	RYR6LT	01:04:48	4.8747	1,000.0000	80.4200	39.0600	370.0000	80.9100	34.2100	360.0000	☑	☐
48	THY4331	RYR6LT	01:10:24	1.9148	1,000.0000	41.0600	38.4500	370.0000	40.3800	36.6600	360.0000	☑	☐
52	THY84U	EXS46Y	01:28:00	2.9767	1,000.0000	-117.1000	-17.2200	370.0000	-114.2300	-16.4300	380.0000	☑	☐
56	EXS76NA	THY1DU	01:43:12	2.3101	1,000.0000	-62.9100	60.8900	380.0000	-62.0400	58.7500	370.0000	☑	☐
60	UAE242	EZY73VA	02:37:36	3.2674	1,000.0000	-113.7300	-18.1800	390.0000	-110.5300	-18.8400	380.0000	☑	☐

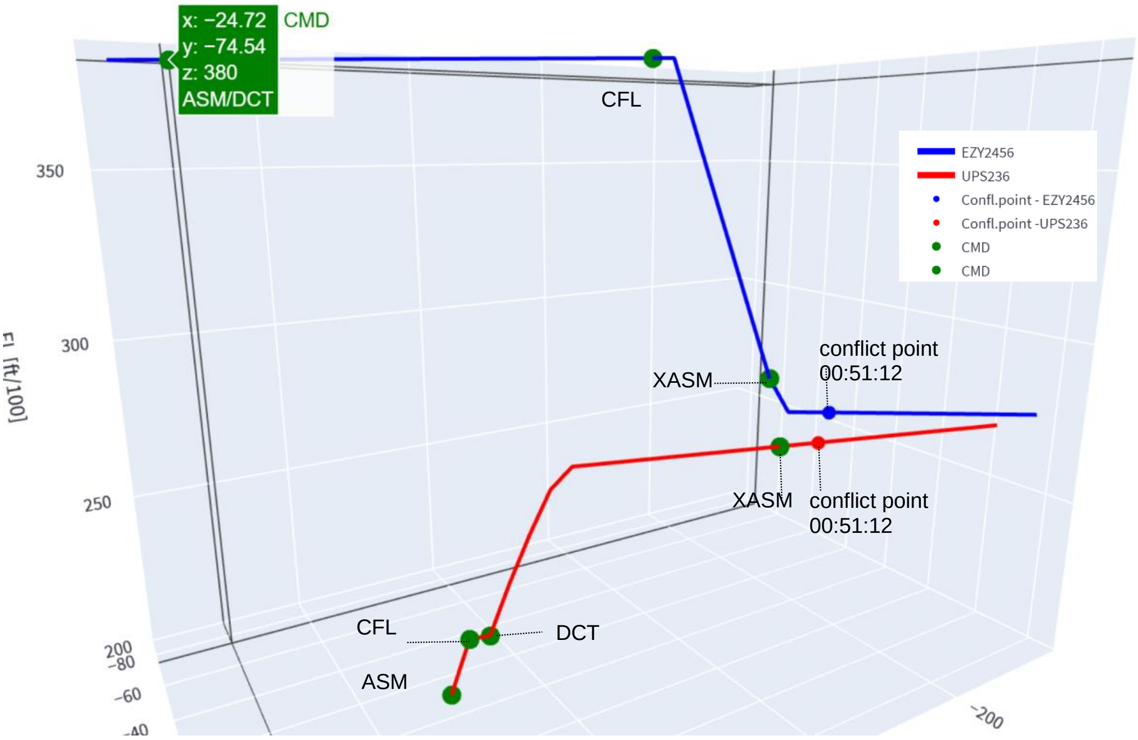


Fig. 12 Trajectory of two flights resulting in a separation infringement

On the other hand, I-Scores at X5 and X10 rates are even lower than the previous I-Scores, as shown in Table 26. For X5, the distinction by the fact that there was only one flight that had inefficient clearances, and only one had a steep turn.

Table 26 I-Score results at X5 and X10 rate

Quantifier	Flights analyzed (X5)	Contributing flights (X5)	Total penalty (X5)	Flights analyzed (X10)	Contributing flights (X10)	Total penalty (X10)
Inefficient clearances	14	1	7.14	17	4	14.71
Inefficient path	151	108	17.27	151	108	17.07
Inefficient climb to ECL	17	2	1.59	19	2	2.15
Horizontal deviation	151	149	12.37	151	147	10.88
Vertical deviation	151	25	6.06	151	34	9.1
Mismatched clearances	151	0	0	151	0	0
Inefficient exit	47	12	3.11	47	34	4.28
Exaggerated Turns	4	1	6.25	5	0	0
Inefficient cruise	47	2	2.55	47	2	2.34
I-Score	56.3			60.6		

4.3 Previous version vs current software version

The statistics derived from the logs of the previous ARGOS version (at X1 rate) and the current version are presented in Table 27.

Table 27 Statistics for the logs of the previous and the current ARGOS version

Statistics	**Previous version**	**Current version**
Total flights	151	151
Flights with complete trajectory	141	151
Simulation time	05:01:50	05:25:31
Nr. CFL instructions	81	92
Nr. DCT instructions	156	159
Nr. HDG instructions	36	5
Average turn (°)	10.9	18
Maximum turn (°)	46	50
Minimum turn (°)	4	5
Nr. DCT combined with CFL	22	14
Nr. DCT combined with HDG	0	0
Nr. CFL combined with HDG	4	3
Avg. time to send first instruction (s)	101.8	11.7
Max. time to send first instruction (s)	485	86
Min. time to send first instruction (s)	9	9
Predominant first instruction	DCT	DCT
Nr. collisions avoided	58	40

The data shows the current version is better in terms of performance. The previous version had much more time until the first instruction was given and issued more clearances to solve conflicts with high turn values.

Surprisingly, D-Score is 99. The only difference compared to the current version at the X1 rate is given by the track leaving boundary metric, with only one contributing flight, as shown in Table 28.

Table 28 D-Score comparison between previous and current prototype versions

Quantifier	**Occurrences (previous)**	**Global penalty (previous)**	**Occurrences (current)**	**Global penalty (current)**
AIRPROX	0	0	0	0
TSA penetration	0	0	0	0
STCA	0	0	1	5
Critical vertical clearance	0	0	0	0
Critical horizontal clearance	0	0	0	0
Critical ICFL	0	0	0	0
Track leaving AoR	0	0	0	0
Mismatched exits	0	0	0	0
Track leaving boundary	1	1	6	6
D-Score	99		94	

The I-Score is 95.9, which is higher by 39% compared to the current version. The penalties are presented in Table 29.

Table 29 I-Score comparison between previous and current prototype versions

Quantifier	**Flights analyzed (prev.)**	**Contributing flights (prev.)**	**Total penalty (prev.)**	**Flights analyzed (current)**	**Contributing flights (current)**	**Total penalty (curr.)**
Inefficient clearances	12	5	41.66	12	1	8.33
Inefficient path	141	80	11.32	151	111	17.58
Inefficient climb to ECL	16	10	3.75	16	3	1.58
Horizontal deviation	141	141	8.31	151	149	13.77
Vertical deviation	141	8	3.37	151	21	6.33
Mismatched clearances	0	0	0	151	0	0
Inefficient exit	42	7	10.79	47	20	4.26
Exaggerated Turns	9	0	13.77	5	2	15
Inefficient cruise	42	1	2.85	47	2	2.61
I-Score	95.9			69.7		

Results show an increase of 55% in the inefficient path, 65% in horizontal deviation, and 87% in vertical deviation for the current version, compared to the previous version. Even if the I-Score has higher values in the earlier version, deviations that already had high values increased meaningfully. Also, there was a minor increase in the exaggerated turns metric. Even so, there was a considerable decrease in frequent clearances, which is five times lower in the current version. Moreover, the software update significantly decreased inefficient climb, cruise, and exit.

CHAPTER 5 CONCLUSIONS AND FURTHER IMPROVEMENTS

Scoring results indicated that the current ARGOS version is better in terms of efficiency. Metrics data revealed that the better overall performance came with a cost: a high increase in criteria already at high levels (deviation) and a slight decrease in safety.

By using the D-Score, it can be said that both system versions can be considered safe, as no significant safety-related problems were detected. But, there are still many questions ARGOS needs to answer, and one of them is related to the incident where a violation of separation was detected below the controlled airspace. Furthermore, no TSA was active during the simulation; thus, the TSA penetration metric couldn't be affected.

It can be stated that the scoring mechanism will contribute directly to the automation in ATC. Firstly, the mechanism can ensure that the automated ATC systems are making accurate decisions, as the algorithm scores the software based on how closely its decisions align with human decision-making standards, reducing the likelihood of errors and increasing safety. Secondly, the scoring mechanism provides an objective way to evaluate the performance of automated systems, and it can assess the system's ability to handle different types of scenarios. Thirdly, the scoring provides ongoing feedback that can help the system continuously improve its performance after each update. Lastly, the mechanism provides a very granular level of evaluation through specific metrics, that provide detailed insights into the performance of the ATC safety-critical systems.

A certain drawback of the scoring mechanism is that it does not use upper limits for certain metrics; these limits can define whether the deviation from the optimum scenario is acceptable or not. For instance, the scoring may flag any situation where an aircraft is deviating significantly from its planned course, instead of just adding values to the scores.

The scoring can be improved in several ways, including refining specific metrics. This can involve adjusting the weighting of existing metrics, or revising the definition of metrics to better reflect their importance to the overall performance of the system. The addition of new metrics can help to improve the fidelity of the evaluation process and can provide more detailed insights into the performance of the system.

A better scoring may rely less on the data provided by ARGOS. To illustrate, the inefficient path metric uses the distance provided by the prototype, which is the Euclidean distance which does not count airspace boundaries. The simulation that ARGOS is currently designed for showed that there were few to no cases when the line between the entry and exit points crossed any airspace boundaries. For upcoming versions and simulation environments, the minimum distance may be calculated using the distance between three points, of which horizontal trajectory would not cross any boundary. To address this question, an algorithm to calculate this distance was provided in APPENDIX C, C5.

The scoring may also be improved if it calculates the TBP independently. Scoring logs have indicated few points when compared to the number of headings. One method to calculate those points is to consider only the flights with more than two headings (or one HDG and after that one DCT), meaning that the first one was given to avoid conflicts while the latter was to resume navigation. The position where the second HDG or DCT was given would constitute the TBP.

Tuning is not only for the prototype but also for the scoring parameters. They may be modified according to a *perfect set of data*, which would be a scenario where the prototype would score excellent; in that case, the I-Score may be tuned to result in a score of 100. Therefore, the 100 I-Score would be a clear reference.

It can be argued that the project’s ambitions were met as it gradually explored the world of ATM/ATC and introduced consistent metrics to measure its performance. It was demonstrated that even a simple scoring algorithm could be implemented for a complex system. The scoring mechanism is not limited to ARGOS, as other automated systems, such as AAP or Skyler, might use it for validation

REFERENCES

[1] FAA. “FAA 711.65 2-1-1”. Archived from the original on June 7, 2010. Available: https://web.archive.org/web/20100607105632/http://www.faa.gov/air_traffic/publications/atpubs/ATC/atc0201.html

[2] Doroteja Timotic, Fedja Netjasov. *Automation in Air Traffic Control: Trust, Teamwork, Resilience, Safety. .* 11th International Conference on Air Transport – INAIR 2022, Returning to the Skies. Available: https://www.sciencedirect.com/science/article/pii/S2352146522006706

[3] Vladimír Socha, Lenka Hanáková, Viktor Valenta, Luboš Socha,, Richard Ábela, Stanislav Kušmírek , Terézia Pilmannová , Jan Tecl. *Workload assessment on air traffic controllers.* 9th International Conference on Air Transport – INAIR 2020, CHALLENGES OF AVIATION DEVELOPMENT. Available: https://www.sciencedirect.com/science/article/pii/S235214652030884X

[4] SESAR Joint Undertaking. *Automation in air traffic management.* Roadmap. (2020). Available: https://www.sesarju.eu/sites/default/files/documents/reports/automation%20atm%20vision%20roadmap.pdf

[5] EUROCONTROL. *About our Maastricht Upper Area Control Centre.* Retrieved from: https://www.eurocontrol.int/info/about-our-maastricht-upper-area-control-centre

[6] EUROCONTROL. Hindsight 33. *MUAC CONOPS 2030.* Available: https://skybrary.aero/sites/default/files/bookshelf/32623.pdf

[7] EUROCONTROL. *MUAC ENVIRONMENTAL REPORT.* 2022. Available: https://www.eurocontrol.int/sites/default/files/2022-02/eurocontrol-muac-environmental-report-2022.pdf

[8] EUROCONTROL MUAC. *ARGOS IN MADAP ORD.*

[9] EUROCONTROL. *EUROCONTROL Maastricht Upper Area Control Centre OPS & Automation Strategy. (*2019). Available: https://skybrary.aero/sites/default/files/bookshelf/5341.pdf

[10] ICAO Doc 4444: Procedures for Air Navigation Services- Air Traffic Management (PANS-ATM), 16th Ed., ICAO: Montreal (2016). Available: https://ops.group/blog/wp-content/uploads/2017/03/ICAO-Doc4444-Pans-Atm-16thEdition-2016-OPSGROUP.pdf

[11] SKYbrary Article: Air Traffic Management (ATM). Available: https://skybrary.aero/articles/air-traffic-management-atm

[12] ICAO Annex 11: Air Traffic Services, 13th edition, July 2001. Available: https://skyrise.aero/wp-content/uploads/2017/03/ICAO-Annex-11-Air-traffic-services.pdf

[13] Boldmethod article: *How To Use An IFR Cruise Clearance*. Available: https://www.boldmethod.com/learn-to-fly/regulations/ifr-cruise-clearance-how-to-fly-with-it/

[14] Steve Reisser, *Air Traffic Control and Clearances* presentation, Instrument Ground School 2015. Available: https://groundschool.weebly.com/uploads/2/3/6/7/2367787/080910_atc_and_clearances.pdf

[15] ZLA Los Angeles ARTCC Article: *Clearance Delivery – VFR Clearances*. Retrieved from: https://laartcc.org/stm/vfr-clearances

[16] FAA. Air Traffic Control, Section 4. ATC Clearances and Aircraft Separation. Available: https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap4_section_4.html

[17] SKYbrary Article: Basic Controller Techniques: Vectoring. Retrieved from: https://www.skybrary.aero/articles/basic-controller-techniques-vectoring

[18] SKYbrary Article: Basic Controller Techniques: Vertical Speed. Retrieved from: https://www.skybrary.aero/articles/basic-controller-techniques-vertical-speed

[19] SKYbrary Article: Basic Controller Techniques: Speed Control. Available: https://www.skybrary.aero/articles/basic-controller-techniques-speed-control

[20] SKYbrary Article: Separation Standards. Retrieved from: https://www.skybrary.aero/articles/separation-standards

[21] CSSI. INC Bussiness Aviation Solutions. Article: Reduced Vertical Separation Minimum (RVSM). Retrieved from: https://www.cssiinc.com/business-aviation-solutions/rvsm

[22] EUROCONTROL. Free route airspace. Retrieved from: https://www.eurocontrol.int/concept/free-route-airspace

[23] César Antonio Nava Gaxiola, *The Future of Free Route in the European Airspace: A study quantifying the cos-benefits and safety-cost of its implementation*, Universitat Politècnica de Catalunya, PhD Thesis, November 2019. Available: https://upcommons.upc.edu/handle/2117/177800

[24] ICAO Doc 9426-AN/924. Air Traffic Services Planning Manual. 1st edition, 1984. Available: https://www.icao.int/EURNAT/Other%20Meetings%20Seminars%20and%20Workshops/Global%20ATFM%20Manual%20Coordination%20Team/1st%20Meeting%20at%20ATC%20Global%202012%20and%20EUROCONTROL%20CFMU/ICAO%20Doc%209426_cons_en.pdf

[25] SKYbrary. Article: Conflict Detection. Retrieved from: https://www.skybrary.aero/articles/conflict-detection

[26] EUROCONTROL. EUROCONTROL Specification for Short Term Conflict Alert. 2019. Available: https://web.archive.org/web/20110612132148/http://www.eurocontrol.int/safety-nets/gallery/content/public/ec_specs/es_STCA-11.pdf

[27] EUROCONTROL. EUROCONTROL Guidance Material for Area Proximity Warning. 2019. Available: https://www.skybrary.aero/sites/default/files/bookshelf/5190.pdf

[28] SKYbrary Article: Medium Term Conflict Detection (MTCD). Retrieved from: https://www.skybrary.aero/articles/medium-term-conflict-detection-mtcd

[29] SKYbrary Article: Controller Workload. Retrieved from: https://skybrary.aero/articles/controller-workload

[30] Italo Romani de Oliveira, Renato Jorge Galvão Teixeira, Paulo Sérgio Cugnasca. (October 2006). *Balancing the air traffic control workload through airspace complexity function*. Conference paper. Available: https://www.researchgate.net/publication/268149387_Balancing_the_air_traffic_control_workload_through_airspace_complexity_function

[31] NASA. *An Evaluation of Air Traffic Control Complexity*. Report of NASA contract NAS2-14284, Advanced Air Transportation Technology (AATT), Program 31. Available: https://www.researchgate.net/publication/268149387_Balancing_the_air_traffic_control_workload_through_airspace_complexity_function

[32] SKYbrary Article: Probe(What-if) Controller Tool. Retrieved from: https://www.skybrary.aero/articles/probe-what-if-controller-tool

[33] In D. De Waard, K. Brookhuis, F. Dehais, C. Weikert, S. Röttger, D. Manzey, S. Biede, F. Reuzeau, and P. Terrier, (Eds.)(2012), Human Factors: a view from an integrative perspective. Proceedings HFES Europe Chapter Conference Toulouse 2012. (ISBN 978-0-945289-44-9) Retrieved from: http://hfes-europe.org.

[34] Allison Leking.IBM. (2022). Co-creating the future of air traffic management. Retrieved from: https://www.ibm.com/case-studies/luftfartsverket/

[35] Sofia Rydell, Martin Christiansson, Magnus Nylin, and Jens Nilsson. (2021). Advanced Autoplanner – A Competent Digital Colleague. LFV – Air Navigation Services of Sweden. Departure of Research and Innovation. Available: https://www.sesarju.eu/sites/default/files/documents/sid/2021/posters/SIDs%202021%20poster%20advancedautoplanner.pdf

[36] AIRTC. (2019). SKYLER. ATC Artificial Intelligence Agent. Skyler CONOPS 1.0. Available: https://www.airtc.io/wp-content/uploads/2019/08/Skyler-CONOPS-_1.0.pdf

[37] EUROCONTROL. EUROCONTROL Specification for Trajectory Prediction. EUROCONTROL-SPEC-0143. (2017). Available: https://www.eurocontrol.int/publication/eurocontrol-specification-trajectory-prediction

[38] SKYbrary Article: AIRPROX. Retrieved from: https://www.skybrary.aero/articles/airprox

[39] IAA. Guidance on Temporary Airspace. Aeronautical Services Advisory Memorandum (ASAM). (2015). Available: https://www.iaa.ie/docs/default-source/publications/advisory-memoranda/aeronautical-services-advisory-memoranda-(asam)/guidance-material-on-temporary-airspace.pdf?sfvrsn=22ad0df3_6

[40] FAA. *Measuring Air Traffic Controller Performance in a High-Fidelity Simulation.* FAA Report: DOT/FAA/AM-00/2. (2000).

[41] Luke L. Jensen, R. John Hansman. Fuel Efficiency Benefits and Implementation Considerations for Cruise Altitude and Speed Optimization in the National Airspace System. MIT. (2014). Available: https://dspace.mit.edu/bitstream/handle/1721.1/88517/ICAT_2014_Jensen.pdf?sequence=1&isAllowed=y

[42] IATA. Single European Sky. Retrieved from: https://www.iata.org/en/about/worldwide/europe/single-european-sky/

[43] SESAR. About SESAR. Retrieved from: https://www.sesarju.eu/discover-sesar

[44] SESAR. European ATM Master Plan. (2020).

[45] ICAO. ICAO Environmental Report 2016. (2016). Pages 138-140. Available: https://www.icao.int/environmental-protection/Documents/ICAO%20Environmental%20Report%202016.pdf

[46] EUROCONTROL. Performance Indicator – Horizontal Flight Efficiency. (2014).

[47] EUROCONTROL. Vertical Flight Efficiency. Performance review commission. (2008). Available: https://www.eurocontrol.int/sites/default/files/2019-05/vertical-flight-efficiency-mar-2008.pdf

[48] Rich Stowell. AOPA Article: All types of turns. Retrieved from: https://www.aopa.org/training-and-safety/students/presolo/skills/all-types-of-turns

[49] ICAO. Rules of the Air. Annex 2 to the Convention on International Civil Aviation. 10th Edition, July 2005. Available: https://www.icao.int/Meetings/anconf12/Document%20Archive/an02_cons%5B1%5D.pdf

[50] ICAO. Operations of Aircraft. Annex 6 to the Convention on International Civil Aviation. 10th Edition, July 2018. Available: http://www.icscc.org.cn/upload/file/20210603/20210603132536_27477.pdf

[51] SKYbrary Article: Basic Controller Techniques: Level change. Retrieved from: https://www.skybrary.aero/articles/basic-controller-techniques-level-change

APPENDIX A. ARGOS OUTPUT DATA

Table 30 shows an example of ARGOS output data containing the instructions given to the flights. The data is of various types, as follows:

- Type – It is the type of action that ARGOS sends or recognizes. In most cases, the type is *CMD*, meaning that it is a command (instruction). Other types are *STCA* (see section 1.5) and SI, which stands for separation infringement – a violation of the separation standards (see 1.2).
- Time – The time when the action was taken.
- CS – Call sign (the identification of aircraft).
- PlanNR – Aircraft identification number. It is taken orderly, i.e., the first aircraft that ARGOS has identified (and assumed in airspace) is 1, the second is 2, etc.
- CMD – The instruction sent to the flight. It can be of type ASM (assumed), meaning that the aircraft was intercepted and ARGOS is now in control of it; DCT (Direct-to); CFL (Cleared Flight Level); HDG (Heading).
- ASMC – its value is relevant only when an STCA alarm is triggered, and it is usually BOL, meaning that the STCA was triggered in the BOL region of the 23 active sectors.
- NCOP – Planned entry waypoint.
- XCOP – Planned exit waypoint.
- exitPT – Actual exit point.
- EFL – Entry flight level
- ECL – En-route Cruising Level, which is the same as the Requested Flight Level (RFL) in the flight plan – if the pilot is cleared to fly RFL [37]. In the other cases, the ECL is the FL maintained from the top of climb (ToC) until top of descent (ToD).
- XFL – Exit flight level.
- ADEP – Departure airport
- ADES – Destination airport.
- CFL – The level at which the flight is cleared to climb/descend (Cleared Flight Level)
- modeC – Altitude.
- pX – Lateral positon.
- pY – Longitudinal position.
- PT – Waypoint. It is filled in combination with a CMD of type DCT, which translates to Direct-to waypoint PT.
- HDG, HDG_REL, HDG_CURR – Are related to the values of the absolute, relative, and current heading.
- CONF_T, CONF_DIST – Only in combination with STCA alarms or SI. They give more information about the conflict.
- CURR_ASM – The number of current assumed (ASM) flights by ARGOS.

Table 30 Example of instruction log file (SCORING)

scoring_date : 2022-08-30_20:04																	
exercise_name : 20220508_2230-0330																	
Type	Time	CS	PlanNR	CMD	ASMC	NCOP	XCOP	exitPT	EFL	ECL	XFL	ADEP	ADES	CFL	modeC	pX	pY
CMD	0:03:16	TRA6506	2	ASM	BOL	EEDER	NORKU		360	360	260	OJAI	EHAM				
CMD	0:04:52	TRA6506	2	DCT								OJAI	EHAM				
CMD	0:05:40	EXS28RY	3	ASM	BOL	BITBU	LUMEN		380	380	380	LGIR	EGCC				
CMD	0:06:28	TFL602	8	ASM	BOL	LUMIL	DENUT		370	370	250	GCFV	EHAM				
CMD	0:07:16	EXS28RY	3	DCT								LGIR	EGCC				
CMD	0:07:16	TFL602	8	DCT								GCFV	EHAM				
CMD	0:08:04	TFL602	8	CFL								GCFV	EHAM	250			
CMD	0:08:04	THY13T	1	ASM	BOL	KOK	MATUG		390	390	390	KMIA	LTFM				
CMD	0:08:52	EXS4WR	4	ASM	BOL	BITBU	LUMEN		380	380	380	LTFE	EGNX				
CMD	0:09:40	TFL41Y	9	ASM	BOL	LUMIL	DENUT		380	380	250	LEPA	EHAM				
CMD	0:10:28	TFL41Y	9	DCT								LEPA	EHAM				
CMD	0:10:28	TFL41Y	9	CFL								LEPA	EHAM	250			
CMD	0:10:28	THY13T	1	DCT								KMIA	LTFM				
CMD	0:10:28	EXS4WR	4	DCT								LTFE	EGNX				
CMD	0:10:28	PGT1176	14	ASM	BOL	REDFA	TIVUN		370	390	390	EGSS	LTFJ				
CMD	0:11:16	TRA6506	2	CFL								OJAI	EHAM	260			
CMD	0:12:04	PGT1176	14	DCT								EGSS	LTFJ				

Table 31 shows an example of the data contained in the coordinates log. In this case, the coordinates are printed every 48 seconds, because ARGOS was set to X10 rate. To be able to read all this set of data, the schema from Table 32 was provided. Each text before a ";" in the log corresponds to an element from the schema.

Table 31 Example of coordinates log file (UPDATE)

```
==========
0:00:04
==========
0:00:52
==========
0:01:40
3;EXS28RY;1000;-19.71;-78.79;-6.04;4.09;380;EXS28RY;B738;380;LGIR;EGCC;;380;438;0.74;0;;;
==========
0:02:28
3;EXS28RY;1000;-24.55;-75.51;-6.04;4.09;380;EXS28RY;B738;380;LGIR;EGCC;;380;438;0.74;0;;;
4;EXS4WR;1000;-10.53;-84.09;-5.88;3.95;380;EXS4WR;B738;380;LTFE;EGNX;;380;425;0.72;0;;;
==========
0:03:16
2;TRA6506;1000;46.5;-0.95;-5.44;4.79;360;TRA6506;B738;360;OJAI;EHAM;;360;435;0.74;0;;;
3;EXS28RY;1000;-29.38;-72.24;-6.04;4.09;380;EXS28RY;B738;380;LGIR;EGCC;;380;438;0.74;0;;;
4;EXS4WR;1000;-15.24;-80.93;-5.88;3.95;380;EXS4WR;B738;380;LTFE;EGNX;;380;425;0.72;0;;;
8;TFL602;1000;-225.99;-48.66;4.21;5.53;370;TFL602;B38M;370;GCFV;EHAM;;370;417;0.71;0;;;
==========
0:04:04
2;TRA6506;1000;42.15;2.89;-5.44;4.79;360;TRA6506;B738;360;OJAI;EHAM;BOL;360;435;0.74;0;;;
3;EXS28RY;1000;-34.22;-68.96;-6.04;4.09;380;EXS28RY;B738;380;LGIR;EGCC;;380;438;0.74;0;;;
4;EXS4WR;1000;-19.94;-77.77;-5.88;3.95;380;EXS4WR;B738;380;LTFE;EGNX;;380;425;0.72;0;;;
8;TFL602;1000;-222.62;-44.24;4.21;5.53;370;TFL602;B38M;370;GCFV;EHAM;;370;417;0.71;0;;;
```

Table 32 Schema list for update log file

trackNR	modeS_CS	modeA	pX	pY	vX	vY	modeC	CS
ACTYPE	CFL	ADEP	ADES	modeS	modeS_GSP	modeS_mach	ROC	

The data stored in Table 31 is similar to the data from the instructions log; thus, the relevant data stored in the update file and hasn't been yet clarified is:

- trackNR – It is the equivalent of PlanNR.
- modeS_CS – Equivalent of CS.
- vX – Longitudinal velocity.
- vY – Lateral velocity.
- ACTYPE – Aircraft type.
- modeS_GSP – Ground speed.
- modeS_mach – Mach number

APPENDIX B. Scoring metrics

Table 33 shows all the metrics along with their penalties.

Table 33 Scoring summary metrics

Quantifier	Scenario	Global penalty	Flight penalty	Flights analysed
AIRPROX	Not applicable	100	Not applicable	All flights
TSA penetration	Not applicable	100	Not applicable	All flights
STCA	Not applicable	5	Not applicable	All flights
Critical horizontal clearance	AIRPROX	10	Not applicable	Flights with TBP
	TSA penetration	10	Not applicable	Flights with TBP
	Sector boundary	5	Not applicable	Flights with TBP
Critical vertical clearance	AIRPROX	10	Not applicable	Flights with CFL
	TSA penetration	10	Not applicable	Flights with CFL
	Sector boundary	5	Not applicable	Flights with CFL
Critical ICFL	AIRPROX	10	Not applicable	Flights with ICFL
	TSA penetration	10	Not applicable	Flights with ICFL
	Sector boundary	5	Not applicable	Flights with ICFL
Track leaving: AoR	Not applicable	20	Not applicable	All flights
Track leaving: boundary	Not applicable	1	Not applicable	All flights
Mismatched exits	Not applicable	20	Not applicable	All flights
Inefficient clearances	2 HDG < 3 min	Not applicable	50	Flights having at least 1 HDG
	2 CFL < 3 min	Not applicable	50	Flights having at least 1 CFL
	> 5 clearances/15min	Not applicable	60	Flights having at least 1 HDG or 1 CFL
Inefficient path	Not applicable	Not applicable	Percentage of exceeded distance (%)	Flights with complete trajectory
Inefficient climb to ECL	Not applicable	Not applicable	Percentage of exceeded distance (%)	Flights that reached ToD
Horizontal deviation	Not applicable	Not applicable	Value of the horizontal deviation (nm)	Flights with complete trajectory
Vertical deviation	Not applicable	Not applicable	Value of the vertical deviation (ft)/50	Flights with complete trajectory
Missmatched clearances	Not applicable	Not applicable	100	Flights with complete trajectory
Inefficient exit	Not applicable	Not applicable	Value of the time spent at TFL (s)/10	Flights with complete trajectory and Flights with ECL>TFL
Exaggerated turns	Turn higher than 20 °	Not applicable	Value of the turn (°)	Flights that have at least 1 turn
	More than 2 turns higher than 15 °	Not applicable	Value of the max. turn (°)	
Inefficient cruise	Not applicable	Not applicable	planned_{ECL} - actual_{ECL}	Flights with complete trajectory

APPENDIX C. EVENT DETECTION LOGIC

The following sections describe the algorithm logic of the function used to calculate important metrics. The functions that are not presented require simple data manipulation.

The scoring app was written in Python 3.9. The necessary data was manipulated with the help of the following packages: *pandas, sys, datetime* and *numpy.* Trajectories were generated using *plotly* while the app interface was created with *streamlit* and *st_aggrid*.

C1. Critical horizontal clearance detection logic

The algorithm is designed to introduce extrapolated coordinates of a possible trajectory after passing the TBP. The function *CHC* takes two inputs, the command data/scoring log *(cmd)* and the coordinates/update log *(crd*).

It filters the command data only to include rows where the value in the *CONF_REASON* column is *TBP passed*. The resulting data is then reset and indexed because only the flights that passed the TBP will have an extrapolated trajectory.

A loop is then executed to iterate through the filtered command data. The aircraft's flight level and time are extracted from the command data in each loop iteration. The coordinates data is then filtered only to include rows where the flight name – call sign *(modeS_CS)* matches the current CS and the time is greater or equal to the current time.

The first row of the filtered coordinate data is then extracted and used to create a new dataframe copy. The new dataframe is modified to contain the same values as the first row but repeated for *cf.CHT + 120* number of times. A new column *seconds* is created in the dataframe and it is initialized to a range of values from *0* to *cf.CHT +120.*

Next, the *Time* column of the copied data is updated by adding the values in the *seconds* column as a timestamp to the original value. The *pX, pY* columns are then updated using the values of the velocity and position at time 0*.* The estimation of the next position is done using forumula:

$$pX = T_{current} \cdot \frac{v_0}{60} + pX_0$$

The *seconds* column is dropped from the dataframe while a new column *Extrapolated* is then added and is set to *CHC.* Finally, the copied dataframe is concatenated with the initial one and the process is repeated for each iteration in the loop.

The function *CHC* only appends new extrapolated values to the initial update log. These new values are then inserted into the AIRPROX function to verify if the fictive coordinates generate AIRPROX.

Note that the extrapolated coordinates follow only a horizontal trajectory, as the condition for a critical vertical clearance is to continue on the same heading after passing the TBP.

```
function CHC(cmd, crd)

  data = get rows from cmd where "CONF_REASON" is equal to "TBP passed"
  for i = 0 to length of data
    fl_name = get value of "CS" at index i in data
    time = get value of "Time" at index i in data

    crddata = get rows from crd where "modeS_CS" is == fl_name  and "Time" is >= (time - 120 seconds)
    px = get value of "pX" at index i in crddata
    py = get value of "pY" at index i in crddata
    vx = get value of "vX" at index i in crddata
    vy = get value of "vy" at index i in crddata

    first_row = get first row of crddata
    crdcopy = repeat first_row (CHT + 120) times
    add column "seconds" in crdcopy with values ranging from 0 to (CHT + 120)
    add column "Time" in crdcopy by adding the value of each row in "seconds" to the corresponding "Time"
    add column "pX" in crdcopy by adding (each row in "seconds" * vX/60 to pX
    add column "pY" in crdcopy by adding (each row in "seconds" * vY/60) to pY
    remove column "seconds" from crdcopy
    add column "Extrapolated" in crdcopy with value "CHC"
    add crdcopy to crd
  end for

  return crd
end function
```

An example of how the extrapolated coordinates are verified in the AIRPROX function is shown in Fig. 13. The aircraft received a first heading, probably to avoid conflicts, and the second one was to redirection it to the RRP point. The purple line is the actual trajectory of the aircraft given in the update log, while the blue dots form the possible trajectory after reaching the TBP. The trajectory after TBP is calculated based on the current aircraft velocity and extrapolated for the next 3 minutes, or as given in the scoring parameters.

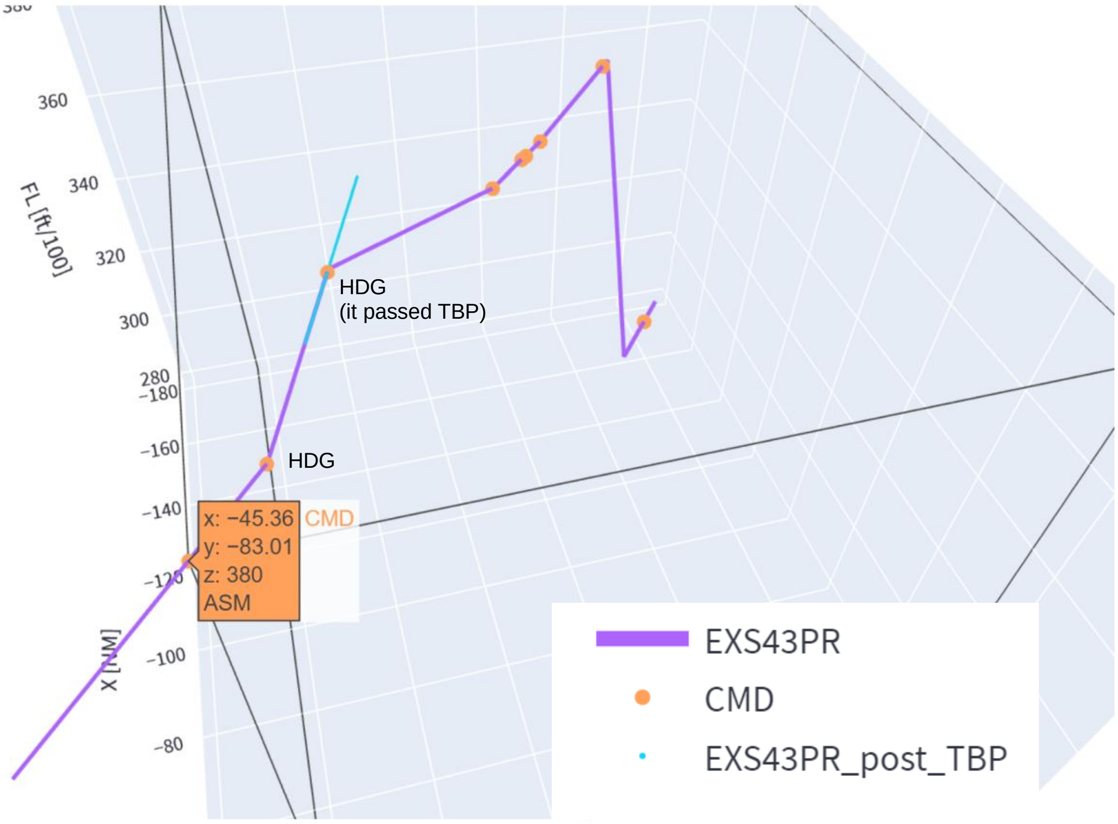


Fig. 13 Example of Trajectory with TBP: EXS43PR

C2. Critical vertical clearance detection logic

The algorithm logic for vertical clearance is the same as for the horizontal clearance: coordinates are generated after each CFL given and appended to the initial update log. The contrast between the two clearances type is that the critical vertical clearance generates multiple possible trajectories after reaching the required FL.

Firstly, the algorithm generates a vertical trajectory using the ROC of the aircraft and velocities, for a 3-minute time after passing the CFL. For each point in the vertical trajectory, new trajectories are generated considering that the flight can level off at those points, until reaching the 3-minute time. The final shape of the trajectory will be a triangle (in 3D space) where each point is associated with its *time* and extrapolated *(x, y, z)* coordinates.

The first part – vertical trajectory is generated using formulas:

$$x_{next} = t_{passed} \cdot \frac{vX_0}{60} + pX_0$$

$$y_{next} = t_{passed} \cdot \frac{vY_0}{60} + pY_0$$

$$z_{next} = t_{passed} \cdot \frac{ROC_0}{60} + \mathrm{mod}\, eC_0 \cdot 100$$

The second part – horizontal trajectory is generated in the same way the critical horizontal clearance is, but for each *modeC*.

The *CVC()* function takes two inputs, *scoringLog(cmd)* and *updateLog(crd)*, and generates trajectories based on *GenVertProfile()* and *HorzProfile()* functions. The algorithm first filters the flights with the CFL instruction from the input *cmd* and then for each flight, it filters the coordinate data for a time window of 20 seconds before the requested time for the CFL instruction. It then finds the points in the coordinate data where the requested CFL is changed and the *modeC* reaches the requested CFL. If the requested CFL is not equal to XFL, it is an intermediate CFL and generates horizontal profiles for each modeC in the filtered coordinate data. For the point where the *modeC* reaches the requested CFL, the function generates a new vertical profile using the *GenVertProfile()* function and a new horizontal profile using the *newHorzProfile()* function. The new vertical and horizontal profiles are then concatenated to the *crd2* dataframe which contains the updated coordinates.

GenVertProfile() outputs a dictionary *vrt_prf* with keys corresponding to the columns names in *crd*. The vertical profile of the aircraft is generated by extarpolating its position and altitude every 5 seconds until either *cf.CVT* seconds have passed or the altitude reaches the *requested_cfl + cf.CVA* or *requested_cfl-cf.CVA* depending on the value of *case* (climb/descend). The position and altitude are calculated using the velocity values and the initial position and altitude of the aircraft. The timestamp is also calculated by adding 5 seconds to the current time at each iteration. The values of unused columns in *crddata2* are set to predefined values and stored in *vrt_prf*. In summary, *GenVertProfile()* function generates a vertical profile for aircraft by extrapolating its position and altitude every 5 seconds until either a maximum amount of time or a maximum altitude is reached.

HorzProfile() takes as input *vertprfdf* and *vx, vy* which are the vertical profile dataframe and velocity in the x and y direction respectively. It loops through each row of *vertprfdf* and then for each row, it extracts the relevant information such as time remaining until reaching the altitude/max. timestamp and positions and stores them in variables *time_remaining*, *px, py.*

It then creates a new copy *crdcopy* by concatenating the current *row* for *time_remaining/5* number of times. The time column is updated to include the seconds elapsed from the original time. The positions are then updated using the formulas from the critical horizontal clearances, using the velocity and position values extracted previously. Finally, the *crdcopy* is concatenated to *horzprf* which was initially the *vertprfdf*. It returns *horzprf*.

```
function CVC(cmd, crd)

  initialize crd2 = crd

  get all cmd with 'CMD' = 'CFL' into cmdcfl_all

  get the unique values of 'CS' column in cmdcfl_all into flights

  for each flight fl in flights
    get the rows in cmdcfl_all where 'CS' = fl and store it in cmdcfl

    for each row i in cmdcfl
      get the values of 'CFL' and 'Time' in row i and store it in requested_cfl and
                                                                requested_time
      get the value of 'modeC' in row i and store it in actual_FL
      if actual_FL is less than requested_cfl
        case = 1
      else
        case = 2
      filter crd to get all rows where 'modeS_CS' == fl
      get all rows where 'Time' >= requested_time - 20s and store it in crddata
      initialize j = 0

      while crddata['CFL'][j] != requested_cfl
        j = j + 1
        if j > length of crddata - 1, break
      if j > length of crddata - 1, break

      first = j
      while crddata['modeC'][j]  != requested_cfl
        j = j + 1
        if j > length of crddata - 1, break
      if j > length of crddata - 1, break

      second = j
      generate the vertical profile using GenVertProfile function and store it in
                                                                        vertprfdf
      generate the horizontal profile using newHorzProfile function and store it
                                                                        in horzprf
      drop 'TimeRemaining' column from horzprf
      concatenate horzprf with crd2
    end for
  end for

  return crd2
end function
```

C3. AIRPROX detection logic

For AIRPROX, only the update log is used. After the data is cleaned and processed into a dataframe, the current update log receives a new column called *Extrapolated,* and the value *Real* is attributed to each element. The column real stands for "real coordinates" provided by ARGOS. It will be used to differentiate between the extrapolated data for critical clearances where the elements will have other attributes, e.g., "CHC" – critical horizontal clearance extrapolation.

The algorithm evaluates the occurrence of the event known as a separation infringement between two aircraft in a given data. It starts by creating four dictionaries, *airprox_data, cvc_data*, *chc_data,* and *icfl_data,* that will store information about the *Flight1, Flight2, Time*, vertical distance *(VDist)*, horizontal distance *(HDist)*, positions *(pX1, pY1, modeC1, pX2, pY2, modeC2),* RVSM conditions *(RVSM* – True or False) and if the airprox was false positive *(Real)*.

It loops through each unique time (outer loop: *for time in timelist*) in the data, and for each time, it selects the rows from the dataframe where *Time* column is equal to the current time and creates a copy of it as *data*.

The inner loop *(for i in range(len(data)-1)* runs over all rows of the *data*, except the last one. For each iteration, the values for *x1, y1, z1* are taken from the *data*'s current row, representing the *pX, pY* and *modeC* columns respectively.

The inner-inner loop *(for j in range(i+1, len(data))* iterates through the remaining rows of the *data*, starting from the row following the current row of the outer loop. The values of *x2, y2, z2* are taken from this row, representing the *pX, pY* and *modeC* columns respectively.

Then, the code checks if the flights represented by the two rows are different, as indicated by the values in the *modeS_CS* column. If they are different, the algorithm checks for an AIRPROX/SI incident.

An SI incident occurs when the horizontal distance between the two flights is less than *cf.*HORSM (config file – horizontal distance), and the vertical distance between them is less than either *cf.RVSM* or *cf.VSM*, depending they are in the same altitude range or not.

If an airprox incident is detected, the values of the columns of the two rows involved are stored in *airprox_data*. Additionally, in a given *metrics_table* dataset, the column *AIRPROX* is set to 1 for both of the flights involved.

The code only considers SI incidents where both flights are "Real" as indicated in column *Extrapolated.* If the flights are not *Real,* it means they were extrapolated for critical clearances. Depending on their extrapolation value, the column corresponding to the type of the critical clearance of the *metrics_table* dataset is set to 1 for both aircraft.

```
for each time in timelist:
    data = get data for the current time from dataframe df
    for each i in 0 to length of data - 1:
        get x1, y1, and z1 for current index i from data
        for each j in i+1 to length of data:
            get x2, y2, and z2 for current index j from data
            if the flight modeS_CS of i and j are different:
                set airprox and rvsm_cond as false
                calculate horizontal distance hdist
                if hdist is less than or equal to cf.HORSM :
                    calculate vertical distance vdist
                    if one of the flights is in a certain cf.RVSM:
                        set rvsm_cond as true
                        if vdist is less than or equal to another constant value:
                            set airprox as true
                    else:
                        if vdist is less than or equal to cf.VSM:
                            set airprox as true
                if airprox is true:
                    set true_airprox as true
                    if rvsm_cond is true and vdist is equal to cf.RVSM:
                        set true_airprox as false
                    if rvsm_cond is false and vdist is equal to cf.VSM:
                        set true_airprox as false
                    if both flights are real, not extrapolated:
                        add values to airprox_data dictionary
                        if true_airprox is true:
                            set "AIRPROX" value in metrics_table to 1 for both flights
```

C4. Inefficient climb to ECL detection logic

The function *IneffClimbToECL()* computes the time spent in airspace until the flight is cleared to ECL. The function takes four inputs: *cmd* (command log), *crd* (coordinates log), *metrics_table,* and *nr_fl_deduct* (number of flights to deduct).

The function starts by initializing an empty dictionary *ineff_climb_data* that will store the flight number, the distance not cleared, the distance to TOD, and the ratio of the distance not cleared to the distance to TOD.

Next, the function filters the flights in *cmd* with *EFL* less than *ECL* and stores the unique callsigns of these flights in *flights*.

For each *flight* in *flights*, the function filters the corresponding rows in *cmd* and *crd* using the *flight's CS* and computes the distance not cleared and distance to TOD. The distance not cleared is the difference of the distances flown between the time when the flight was cleared to ECL and the time when the TOD was reached. The distance to TOD is the sum of the distances flown from the aircraft's start of the climb to the TOD.

Finally, the function computes the *ratio* of the distance not cleared to the distance to TOD and stores all the information in the *ineff_climb_data*. The function returns the *ineff_climb_data*.

```
function IneffClimbToECL(cmd, crd, metrics_table, nr_fl_deduct)

    ineff_climb_data = {"Flight":[], "DistanceNotCleared":[], "DistanceToTOD":[],
                                                                "Ratio":[]}
    flights = unique flight numbers in cmd where EFL < ECL
    count_flights = 0
    for each flight fl in flights:
        filtcmd = cmd records where flight number is fl
        efl = EFL value of filtcmd
        ecl = ECL value of filtcmd

        filtcrd = crd records where modeS_CS is fl
        if modeC of filtcrd > efl:
            d = 0
        else:

            filtcrd_tod = filtcrd records where modeC >= ecl, copy (with deep copy)
            if filtcrd_tod is not empty:
                time_tod = time of last record in filtcrd_tod
                try to get time_cfl from cmd where CFL == ecl and flight number is fl
                asm_time = time of ASM command in filtcmd
                index_cfl = index of first record in filtcrd_up_to_TOD where Time >=
                                                    time_cfl and Time <= time_tod
                convert altitude from feet to nautical miles
                for each record in filtcrd_up_to_TOD:
                    calculate distance to next record
                distance_tod = sum of all distances in filtcrd_up_to_TOD
                distance_cfl = sum of all distances in filtcrd_up_to_TOD where Time
                                                >= asm_time and Time <= time_cfl
                add fl to Flight list in ineff_climb_data
                add distance_cfl to DistanceNotCleared list in ineff_climb_data
                add distance_tod to DistanceToTOD list in ineff_climb_data
                add distance_cfl / distance_tod to Ratio list in ineff_climb_data

                increment count_flights by 1
    update metrics_table by adding count_flights to nr_fl_deduct
    return ineff_climb_data

end function
```

C5. Shortest distance algorithm

ARGOS provides in the output logs the minimum distance between the entry and the exit point. That distance does not consider MUAC's boundaries. For instance, in Fig. 14, the segment [A;B] crosses MUAC's borders; the minimum distance calculated by ARGOS in the output files is the length of the segment [A;B].

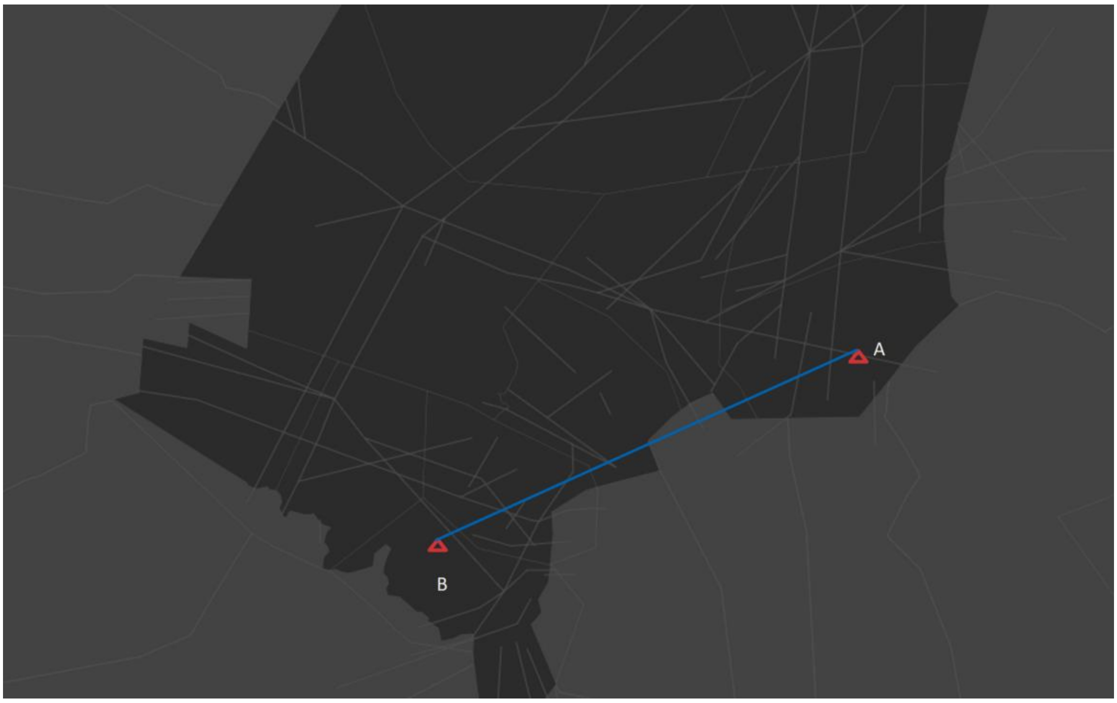


Fig. 14 Euclidean distance between two points in MUAC's airspace

For accuracy, the shortest distance between 2 points that cross the border should be composed of 2 segments. The new distance in this case is calculated as the sum of the distances between 3 points, as show in Fig. 15. Point D represents a threshold because the flight is not allowed to fly exactly above the border. Hence, to determine the minimum distance, the following assumptions are made:

- the new trajectory must be composed of max. 2 lines (equivalent to max. 1 HDG);
- only the horizontal trajectory is considered.

Fig. 15 Accurate distance in airspace

The following steps are required:

1. *Verify the need of this algorithm.*

Check if the line between A and B crosses the airspace near MUAC. This can be done in various ways in Python; the easiest way would be to generate multiple points between A and B that satisfy the linear equation and introduce those points in a pre-built library that checks if a point is inside a polygon.

2. *Generate linear equations for parallel lines with respect to [A;B].*

Generate multiple parallel linear equations with respect to the current line and apply step 1). Stop the algorithm when a parallell line that is completely inside MUAC's airspace was found.

3. *Find possible C points.*

After finding the parallel line that satisfies the condition, go back to the previous generated parallel; the intersection with MUAC's border should be one point. That point is the 'highest point' and its position is considered to be the turning point for the aircraft (see Fig. 16).

To find the C point, the datasets containing the parallel line at which the algorithm stopped, and the previous parallel must be compared. The comparison in those datasets will results in a set of points *that may be* point C – those should be again saved in another dataset.

4. Find point D.

For each C point in the new dataset, a D point will be generated until segments [A;D] and [D;B] satisfy the condition of being completely inside MUAC's airspace.

Fig. 16 Parallel line to [A;B]

From the point C, a perpendicular line with respect to the found parallel shall be calculated. This perpendicular line will contain point D, whereas the distance [D;C] is a given minimum, similar to the VSM (Fig. 17).

Fig. 17 Given minimum to find point D

The minimum distance in this case will be [A;D] + [D;B] instead of [A;B] with D being a point in the airspace where the aircraft should be able to execute a HDG (Fig. 18).

Fig. 18 New trajectory [A;D], [D;B]

Mathematical equations:

Given the coordinates for A(x_1, y_1) and B(x_2, y_2) it can be easily deduced that the line equation is:

$$y - y_1 = m \cdot (x - x_1)$$

whereas $m = \frac{y_2 - y_1}{x_2 - x_1}$ is the slope intercept. The equation linear equation will be:

$$y = m \cdot x + b + c$$

A *true* equivalent point on a parallel line for a given distance between lines is:

$$A'\left(\frac{bp-(b+dist)}{m+\frac{1}{m}}, m \cdot \frac{bp-(b+dist)}{m+\frac{1}{m}} + (b+dist) \right)$$ with $bp = y_A + \frac{1}{m \cdot x_A}$ and $A(x_A, y_A)$.

In Fig. 19, A’ is the true equivalent point while A’’ is the equivalent point.
An equivalent point on a parallel line is $A'' = \left(x_A, m \cdot x_A + b + c\right)$ whereas *c* comes from the equation of a parallel line.

Input:

- (x,y) coordinates for points A,B
- Airspace limits (polygon limits)
- If used independent of the scoring -> area to plot (airspace)

Output:

- Position of point D

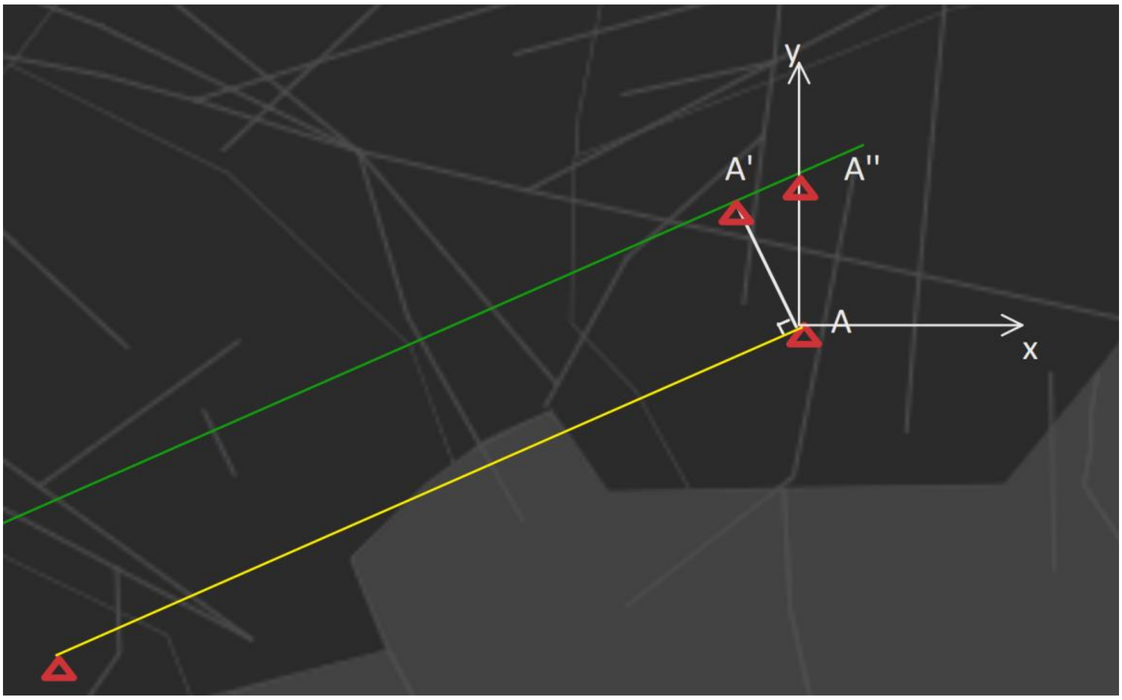


Fig. 19 Different equivalent points wth respect to A and line [A;B]

APPENDIX D. ARCHITECTURAL DESIGN

The scoring app is enabled by the usage of the mentioned packages at the beginning of APPENDIX C. After writing *streamlit run main.py*, the app opens a new page in the browser (working on localhost). The primary file of the application is *main.py* which calls all the functions in the other files that are stored in *classes* folder. The diagram of the app is shown in Fig. 20.

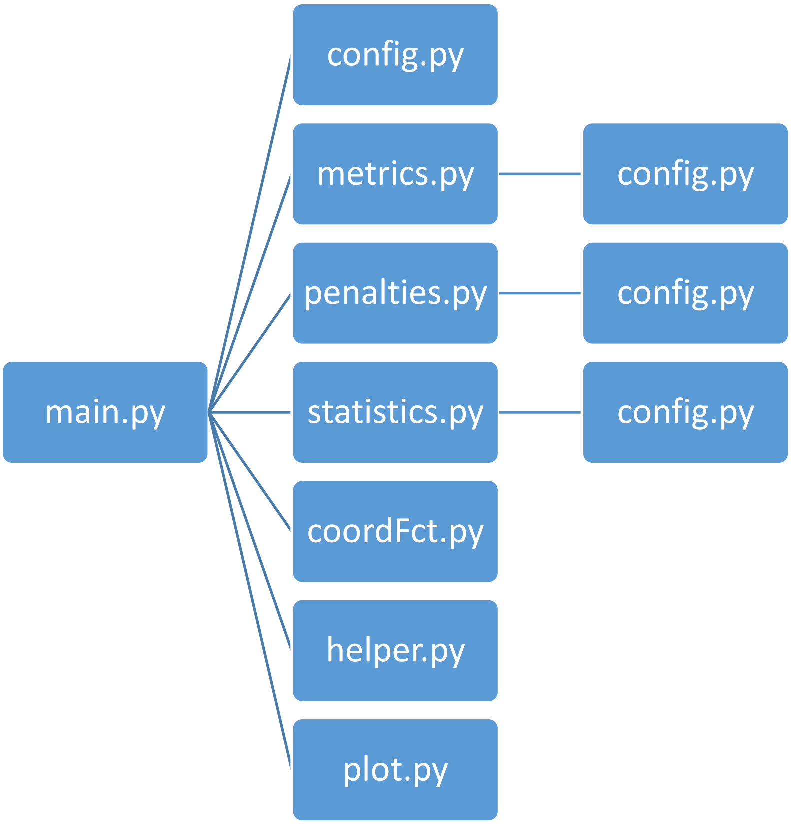


Fig. 20 Scoring app graph

Each script contains the following functions:

main.py
- *Simulation()*
- *load()*
- *gridOpt()*

metrics.py
- *STCA()*
- *SI()*
- *IneffCl()*
  - *SearchClearance()*
- *Deviation()*
- *AIRPROX()*
- *CVC()*
  - *GenVertProfie()*

newHorzProfile()
TrackLeavingBoundary()
ExaggeratedTurns()
IneffPath()
IneffClimbToECL()
Missmatched()
IneffExit()
IneffCruise()
CHC()

penalties.py
InitNrFlightEnteringPenalties()
GlobalPenalties()
FlightPenalties()
ConsidPenalties()

statistics.py
CollisionsAvoided()
Statistics()

coordFct.py
ImportCoordinates()

helper.py
updateCMDinCOORD()

plot.py
make_tracke_sectors()
plotFlight()
plotFlightCVC()
plotFlightCHC()
plotAIRPROX()

The outline of each function is provided in Table 34.

Table 34 Brief description of functions used

Script	Function	Description
main.py	*Simulation()*	Initiates the interface of the app using *streamlit.*The scoring log is read. The update log is read and converted using *ImportCoordinates().* Creates a *metrics-_table* based on the metrics given in *config.py.* Calls all the functions from *metrics.py* to process the created *metrics_table*. After that, *metrics_table* is sent to the functions from *penalties.py* to be processed. All the calculated metrics and penalties are saved and then sent to *load()*
	load()	It is not exactly a function. *load()* is used to create a checkbox for the user, so that when it is ticked, the *load()* receives all the data from *Simulation().* The load

main.py		prevents the page created on localhost to refresh at every interaction. Visual tables, graph and mostly all interaction with the user is created in this page.
	gridOpt()	It stands for grid options. Based on a given table where the first column is a flight CS, it creates an interactive table. By clicking on the Flight name, it creates a plot with the trajectory and displays all the relevant information about that flight.
metrics.py	*STCA()*	Determines the occurences of STCA events based on the scoring log. Each STCA found is deducted in the *metrics_table*. Moreover, it iterates over the time of the found STCAs to determined if it wasn't actually a single event.
	SI()	Counts the number of SI provided in the scoring log and updates the *metrics_table.*
	IneffCl()	Calls function *SearchClearance()* 3 times based on the instructions (CFL, HDG or both) to search for inefficient clearances.
	SearchClearance()	For each clearance type given, it iterates over all flights and for each clearance given it calculates the time between the first and the last clearance. If the time is less than a constant value and there were more than a constant value clearances, the *metrics_table* is updated accordingly.
	Deviation()	Filters flights so that only flights with complete trajectory are analysed. Based on given coordinates in a json file, it searches for the time the flight exited the airspace and calculates the horizontal and vertical deviation. Updates the *metrics_table* accordingly.
	AIRPROX()	Inputs the coordinates/extrapolated coordinates and searches for AIRPROX event using the algorithm provided in previous sections.
	CVC()	Searches for critical vertical clearance by generating vertical trajectories (using *GenVertProfile())* and horizontal trajectories (using *newHorzProfile())*. The extrapolated coordinates are concatenated with the update log.

metrics.py	*GenVertProfile()*	Generates vertical coordinates for a given flight.
	newHorzProfile()	Generates horizontal coordinates for a given flight.
	TrackLeavingBoundary()	In the update log, it searches for data where the *CONF_REASON* column is equal to *Boundary*. It updates the *metrics_table* with value 1 for each flight found at boundary.
	IneffPath()	It calculates for each flight the actual distance flown from the time the aircraft was detected by radar and until it exitted the airspace. The distance is then compared with the *MIN_DIST* provided by ARGOS in the scoring log. If the ratio is higher than a constant, the ratio of the overflown distance is attributed to each flight in the *metrics_table*.
	IneffClimbToECL()	Determines the exceeded ratio of the time the flight was not cleared to ECL. If the ratio is higher than a constant, the *metrics_table* is updated with that ratio for each flight considered.
	Missmatched()	From the scoring log, firstly it searches for the actual exit point of the aircraft and then it compares it to the exit point given in the flight plan (first instruction – when flight was assumed). If they are different, the *metrics_table* is updated with 1 for each flight with wrong exit points. Secondly, it searches for DCT instructions given to other points than the exit point.
	IneffExit()	For each flight that had ECL>XFL, it calculates the time spent at XFL (after climbing). If that time is higher than a constant, *metrics_table* is updated with the time spent at XFL (in seconds).
	IneffCruise()	Using the ECL provided in the scoring log, for each flight it searches if any of the *modeC* values from the update log is higher or equal to ECL. If not, the distance between *modeC.max()* is updated fo each flight in the table.
	CHC()	For each flight containing a TBP, it generates horizontal trajectories after passing TBP until a constant time values is reached. The extrapolated coordinates are then concatenated with the initial update log.

penalties.py	*InitNrFlightEntering .. Penalties()*	It initiates *nr_fl_deduct* dataframe that contains the metrics name. For each metric is calculated the number of flight participating.
	GlobalPenalties()	Using data from *metrics_table*, it creates a new dataframe that contains the penalties applied to each metric and flight in part. The penalty values are taken from *config.py*.
	FlightPenalties()	It follows the same principle as the previous function, the only difference is made by the division of the nr. flights participating in each metric.
statistics.py	*CollisionsAvoided()*	Using only the data from the scoring log, it counts the number of *CONF_REASON* occurences.
	Statistics()	Generates a dictionary with all the necessary statistics that may be analysed using the scoring log.
coordFct.py	*ImportCoordinates()*	Transforms the update log into a readable dataframe using the schema list from APPENDIX A.
helper.py	*updateCMDinCOORD()*	From the scoring log and the modified update log (which contains extrapolated coordinates for CVC and CHC) it creates a new uniform dataset.
plot.py	*make_trace_sectors()*	Using a set of sectors given in a dictionary, it creates a Scatter3D trace of that sector.
	plotFlight()	It uses the name of a flight, scoring/update log to generate a 3D trajectory over the airspace sectors.
	plotFlightCVC()	Uses the same principles as plotFlight() but it adds the extrapolated coordinates for critical vertical clearance.
	plotFlightCHC()	Same as *plotFlightCVC()* but for horizontal clearances.
	plotAIRPROX()	Using data from *airprox_data*, scoring log, update log, and the name of 2 flights it returns an object figure that contains the data to plot 3D the SI event.

The parameters used for each metric in the scoring are as follows:

AIRPROX:

- *RVSM = 2000; % (ft)*
- *VSM = 1000; % (ft)*
- *FLRVSM_MAX = 410; % maximum FL for RVSM*

- *FLVRSM_MIN = 290; % minimum FL for RVSM*
- *HORSM = 5; % (nm) horizontal separation minimum*

STCA:

- *ONOFFSTCA = 10; % (seconds) – multiple STCAs are considered a single event if the OFF/idle time is less than ONOFFSTCA*

Critical horizontal clearance:

- *CHT = 3*60; % Continuing on the current heading for the next CHT seconds causes airprox, c.*

Critical vertical clearance:

- *CVA = 2000; %(ft) - critical vertical altitude (exceeded altitude after reaching requested FL)*
- *CVT = 3*60; % (seconds) – critical vertical time (the time in which conflicts can happen after reaching the requested FL)*
- *CVS = 100; % (ft) – critical vertical step (step used to simulate possible trajectories up to CVA)*

Inefficient clearances:

- *NRGDG = 2; % (nr.) headings in less than TIMEHDG*
- *TIMEHDG = 3*50; (seconds) - NRHDG headings in less than TIMEHDG*
- *NRCFL = 2; % NRCFL CFLs in less than TIMECFL*
- *TIMECFL = 3*60; (seconds)*
- *NRCL = 5; % (nr.) – more than NRCL clearances per TIMECL flying time*
- *TIMECL = 15*60; % (seconds)*

Inefficient path:

- *MINDISTP = 5; % (%) – MINimum DISTance Percentage threshold (ratio actual_distance/minimum_distance > MINDISTP, a penalty is applied).*
- *MINDIST_RADAR = False; % (False if the computed distance is from the point that the aircraft appears on radar and True if the computed distance is from the entry point in flight plan (or at least close).*

Inefficient climb to ECL:

- *ECL_P = 3; % (%) – more than ECL_P % of the flown distance prior to ToD not cleared to ECL.*

Horizontal deviation:

- *HRZDEV = 2; % (nm) – Horizontal deviation threshold.*

Vertical deviation:

- *VRDEV = 200; %(ft) – Vertical deviation threshold.*

inefficient exit:

- *IEXT = 30; % (s) – If more than IEXT seconds are spent at TFL, a penalty proportional to the time spent is applied.*

Exaggerated turns:

- *MDET = 20; % (*°*)* – Minimum Degree to be considered Exaggerated Turn
- *MDET_COUPLE = 15; % (*°*)* – Minimum degree for multiple turns
- *MTE_COUPLE = 2; % (*nr.*) – Minimum turns to be considered exaggerated*
- *MTET_COUPLE = 3*60; % (seconds) – Maximum time betwen MTET_COUPLE turns of at least MDET_COUPLE* degrees to be considerd exaggerated.