Thesis Report

Aircraft conflict detection methods: a data-driven performance assessment based on look-ahead time

Luna Julião

Department of Control & Simulation
Faculty of Aerospace Engineering

Aircraft conflict detection methods: a data-driven performance assessment based on look-ahead time

by

Luna Julião

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday May 13, 2022 at 14:00.

Student number:	5145570	
Project duration:	February 5, 2021 – May 13, 2022	
Thesis committee:	Dr. Ir. J. Ellerbroek,	TU Delft, chair
	Prof. Dr. Ir. J.M. Hoekstra,	TU Delft, supervisor
	Ir. P.C. Rolling,	TU Delft, external examiner

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Cover Image: https://www.internationalairportreview.com/article/98668/nats-apac-air-traffic-management/

TU Delft

Preface

The airspace's demand is continuously increasing, pushing for new records every year. The aviation sector has been exploring new tools and trying to increase the airspace capacity to keep up with the foreseen high demand levels. This research aims to contribute to the knowledge of the accuracy of existing conflict detection methods as a function of the look-ahead time, by aiming for realistic simulation conditions using recorded air flight data, different traffic densities, and meteorological conditions. This project was carried out always with a special concern to make it available for possible future research, thus the code is made available for the entire ATM community.

This project is the culmination of my MSc studies in Aerospace Engineering, it's the closure of an adventure and an important chapter in my life. I would like to thank first my supervisor Prof. Dr. Ir. Jacco Hoekstra who contributed with crucial and insightful discussions to this research. Thank you for challenging me and supporting my research decisions, as well as the wise advice for the future that awaits. I have spent two atypical years in Delft, but I am more than grateful for the experiences, the people, and the knowledge I integrated into my life. I have my parents to thank for this opportunity and for teaching me to always pursue new challenges. To all the people that were there through the rough but also bravo moments, a heartful thank you for being part of this journey.

Luna Julião
Delft, May 13, 2022

Contents

Acronyms

ACAS Aircraft Collision Avoidance System. 3

ADS-B Automatic dependent surveillance-broadcast. 37, 41

ATAG Air Transport Action Group. 3

ATC Air Traffic Control. 3, 4, 37

ATM Air Traffic Management. 35, 36

FAA Federal Aviation Administration. 3

FN False Negative. 51, 59

FP False Positive. 51, 59

GA Genetic Algorithm. 47

ICAO International Civil Aircraft Organization. 3

LoS Loss of Separation. ix, 35, 55, 59

TP True Positive. 51, 56, 59, 60

UAV Unmanned Air Vehicle. 3

List of Figures

List of Tables

I

Introduction

1

Research Introduction

1.1. Background

In the aviation sector, a growing concern focuses on the balance between airspace capacity and demand [3]. Air transportation demand is increasing every year and this growth is not expected to slow down anytime soon. Air Transport Action Group's (ATAG) predictions in 2012 foresaw an increment to almost double of commercial aircraft movements by 2030 [1]. Nevertheless, the airspace is limited and its available capacity is decreasing, leading potentially to more conflicts, an excessive workload for the Air Traffic Control (ATC) ground controllers, an increase in costs for airlines, less environmental compliance, among other unenthusiastic factors.

There are pointed out entities such as Federal Aviation Administration (FAA), International Civil Aircraft Organization (ICAO) and Eurocontrol which make their mission solving this issue. There is continuous work on finding new optimal traffic solutions, building new tools, safer guidelines, among others. As an example, FAA developed a new tool named ACAS X, still in validation, that warns the pilot of potential conflicts with a diversified set of air traffic vehicles. This feature is becoming crucial nowadays due to the uncontrolled increment of unmanned air vehicles (UAVs) flying in the airspace.

Due to its importance, potential solutions are continuously explored by specialists that aim to increase airspace capacity by creating, modeling and simulating different conflict detection and resolution methods, trying to transform them into the most efficient and autonomous solutions. This research, however, evolves at a slow pace due to the several broad topics and influencing parameters that make up the whole concept (e.g. trajectory prediction, conflict detection, and conflict resolution)

This research contributes to the overall solution by tackling the conflict detection branch. It will focus on evaluating the performance of existing conflict detection methods as a function of look-ahead time and document the different steps and step-backs to do so. The goal is to bring this experiment as close as possible to what would be the real-life analysis, thus, the work will make use of recorded air traffic data and make only the necessary assumptions. In addition, some classification on the traffic density and meteorological conditions is integrated for additional analysis. The conflict detection methods will be simulated in the air traffic management BlueSky simulator [2] and a detailed analysis on the performance for different look-ahead times will be done.

1.2. Research Objective & Questions

Research Objective

The research objective for this project is presented below.

> "To contribute to the comprehension of Air Traffic Management tools' accuracy by conducting an analysis on different scenarios where conflict detection methods performance is assessed and correlated with the set look-ahead time."

This objective includes some secondary objectives: (1) obtain the most realistic air traffic scenarios by using recorded air traffic data and counteracting the impact of human intervention and (2) understand the impact different inherent scenario characteristics have on the performance.

Research Questions

The final goal of this research is to be able to answer the following question:

"What is the impact different look-ahead times have in conflict detection performance?"

This research aims to answer this specific research question, a culmination of all the sub-questions (SQ) mentioned below.

SQ.1 What are the characteristics of the most realistic (assumptions free) air traffic scenarios?

1.1 What data sources and databases are fit to construct these scenarios?

1.2 How to counteract the impact of human intervention in the air traffic data?

SQ.2 Which metrics and parameters to use when calculating performance of conflict detection methods?

SQ.3 What is the global performance of BlueSky conflict detection methods considering different look-ahead times?

3.1 How does the performance change for different traffic density scenarios and different look-ahead times?

3.2 How does the performance change for scenarios with different meteorological conditions and different look-ahead times?

3.3 What is the conflict detection performance over different flight phases?

3.4 Is it possible to identify a trend in the results? If yes, what is it?

The first sub-question focuses on a crucial primary element, pairing with the first secondary objective. Since one of the objectives is to do this research with the most realistic data, the fittest databases should be used. However, recorded data includes ATC controller's and pilot's intervention, and considering the goal is to evaluate the conflict detection methods, there should be a normal distribution of conflicts as the controller would see it live. However, the recorded data is a record of the past reality when ATC controllers had to intervene and resolve conflicts. Therefore, the initial research will focus on how to solve this necessary step.

It is necessary to understand how the conflict detection performance should be assessed so that the results are indeed useful for future research, keeping the most realistic factors (second secondary objective). Sub-question two aims to bring awareness to the fact the metrics should be determined taking into consideration what we are assessing (look-ahead time) and what the goal of ATC controllers is.

The final sub-question is the one that brings the work closer to answering the main research question. At this phase, the questions focus on the different results that will be brought together to answer the final question, such as the different scenarios and conflict distribution assessments. The evaluation still aims to maintain realism, thus, the chosen assessments were not limiting but, in contrast, representative of what happens realistically (different traffic density, distinct distribution of conflicts for different flight phases).

1.3. Structure

This research encompasses different topics within the Air Traffic Management environment. Initially, Chapter I introduces the growth of this environment and the research design choices according to it.

In Chapter II, two scientific papers are presented. The first element, *Mitigate ATCO Separation Action Bias in Recorded Air Traffic Data with Time Shifting Methods*, focuses on answering the first research questions on eliminating the human bias in recorded air traffic data. The second element, *State-based Conflict Detection Performance as a Function of Look-ahead Time*, compiles all topics of this research: briefly by answering the same questions as the first one, by elaborating a scenario classification based on traffic density and meteorological conditions, and by obtaining the conclusions of the final research questions.

This research's work is wholly described in Chapter II, but some processes and details are not mentioned. Thus, in Chapter III, different subjects are approached such as the *Data Processing* reasoning, the genetic algorithm implementation and validation, and the flight phase identification and influence in the final results.

In the final chapter IV, some conclusions and remarks for improvements on the current research are disclosed and some ideas of future work are suggested.

II

Scientific Articles

2

Mitigate ATCO Separation Action Bias in Recorded Air Traffic Data with Time shifting Methods

Mitigate ATCO Separation Action Bias in Recorded Air Traffic Data with Time Shifting Methods

Luna Julião

Abstract—**This paper focuses on the mitigation of the human intervention bias in recorded air flight data. The knowledge on how to differentiate the human intervention, such as pilot actions and ATC clearances, and the natural flight deviations is crucial for air traffic control tools and conflict detection software development. This research work is based on a previous study completed by authors Paglione, Oaks, and Summerill where the time shifting concept, applied to scenarios built with recorded air traffic data, is introduced and three different time shifting methods were investigated: *random time adjustment*, *time compression*, and *genetic algorithm implementation*.**

This investigation's first step was on data acquisition from Eurocontrol and OpenSky databases, followed by scenario simulations run in the open-source BlueSky simulator. The main step focused on the reproduction and analysis of the different methods, paired with a highlight on the differences between studies. The main differences focus on the simulator used, the nominal scenario data type (filed data vs. actual data), the implementation of the genetic algorithm, and the criteria to select the best time shifting method.

The results showed a significant difference between the actual data and filed data, and a logical step based on the goal to obtain data not affected by the human bias led to the choice of filed data as the nominal values.

The *random time adjustment* and the *implementation of a genetic algorithm* methods are not deterministic, opposed to the *time compression* technique. The results show the more effective method is the implementation of a genetic algorithm, followed by the trend of random time adjustment results and time compression, in that order. These methods showed themselves fit to approximate all variables to the nominal values, except for the encounter geometry variables. Thus, all techniques are evaluated under the same conditions, and even though the *genetic algorithm implementation* is the more effective technique, the random time shift adjustment and time compression techniques require considerably lower computational power.

Index Terms—**Simulation, BlueSky, Conflicts, Time shift, Genetic Algorithm, Recorded air traffic data, Air traffic control, Human bias**

I. Introduction

In the aviation sector, a growing concern focuses on the balance between airspace capacity and demand (2006, [6]). Air transportation demand is increasing every year and this growth is not expected to slow down anytime soon (2021, [9]). Nevertheless, the airspace is limited, and for the last years, its available capacity is decreasing, leading to potential conflicts, an excessive workload for the Air Traffic Control (ATC) ground controllers, an increase in costs for airlines, and less environmental compliance, among other unenthusiastic factors.

In order to surpass this problem, researchers have been studying, simulating, modeling, and generating tools that aim to improve air traffic safety by assisting human controllers in overviewing the air traffic paradigm. Among these authors, some focused on recorded air traffic data and how this could be used to increase the accuracy of the different experiments. However, this method brings a considerable bias included in the data caused by human intervention (pilot decisions and ATC indications). Thus, some of the topics developed around this subject focused on tuning down this human influence, as will be further explored, so that the advantage of using recorded data could still be worth compared to other less reliable approaches.

The work on this subject dates back to the decade 1990 but it's still an ongoing open problem since there are new variables and factors to take into account and space to improve the effectiveness and efficiency of these approaches.

On the one hand, one of the first mechanisms to eliminate this bias was brought by Niedringhaus and Paielli, in 1998. Niedringhaus (1998, [11]) overlayed air traffic data from different flight levels to generate conflicts, in this case limiting the research to leveled flights and taking into account the possibility of inconsistencies from manipulating the data this way. To minimize the potential effect of the wind error, only aircraft with an altitude shift up to 5000ft are taken into account. A different method was used by Paielli (1998, [15]) who did not generate conflicts but altered the required separation, inducing the system to detect conflicts for different conflict detection metrics (up to 9NM). [1]

On the other hand, time shifting is an approach introduced also by authors Alam et al. (2007, [1]), who mentioned the previous strategies (1998, [11], [15]). This approach was developed by Alam et al. employing a genetic algorithm, but Paglione, Oaks, and Summerill (2003, [14]) looked into the time shifting technique and developed it by coming up with three different methods that aimed at generating conflicts with a real-life representative distribution while also minimizing the individual time shift of each flight. The three different methods were: *random time adjustment*, *time compression*, and the *implementation of a genetic algorithm* to obtain the fitter time shifts.

II. Problem Definition

A. Research Objective

The goal of this research is to improve Air Traffic Management tools' accuracy by conducting an analysis on how to mitigate the bias due to human intervention in recorded air flight traffic data.

B. Background Theory

This research makes use of fast-time air traffic simulation to achieve its objective. Air traffic simulators, focusing on the ones used by ATC ground controllers, incorporate conflict detection and resolution features. A key concept for this research is explained below, along with a summary of the *genetic algorithm* concept due to its importance in the research unfolding.

The basic concept that requires comprehension is *Loss of Separation* (LoS), that happens when an aircraft crosses another aircraft's Protected Zone (PZ). On the one hand, as established by ICAO, when using surveillance systems [7], the PZ of an aircraft is, in most cases, 5 nautical miles (NM) horizontally and 1000ft vertically, as illustrated in Figure 1. On the other hand, an LoS is a concept that can be used for other purposes, since it has arbitrary parameters. For this research, an aircraft's PZ is defined with a maximum horizontal distance of 25NM and a maximum vertical distance of 5000ft. This definition includes non dangerous encounters, encounters that require Air Traffic Controller Officer (ATCO) attention and safety critical LoS.

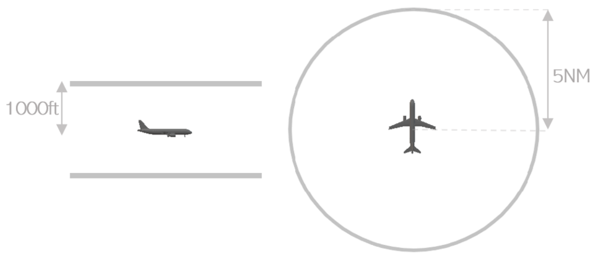


Fig. 1. Protected Zone representation

The genetic theory was first introduced by Holland (1992, [8]). Genetic algorithms (GA) get their fundamental principles from biological phenomena and Charles Darwin's concept of natural selection. Natural selection is a natural process and results in the selection, within populations, of the fittest individuals to a specific environment. The migration into computational algorithms happens through a fitness function where the input elements with a higher fitness function output are selected. The basic GA key elements are clarified below, along with Figure 2 which exemplifies them according to how they are used in this project.

- Individual / Chromosome: An individual or a chromosome, for the specific case both can be considered the same, are the elements of a population. In GA, each one is a potential solution to the problem in question.
- Population: A group of individuals that will reproduce the next generation and from where the fittest elements are selected.
- Gene: A gene is the element of the individual/chromosome, thus always correlated with a specific parameter of the problem's solution.

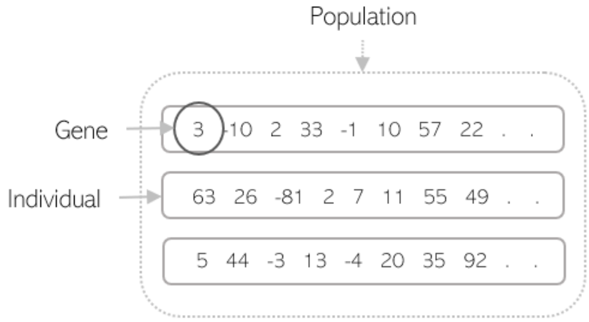


Fig. 2. Genetic Algorithm basic elements

Genetic Algorithms are stochastic algorithms with no guarantee of a good solution, that are iterated until a stop criterion is met (e.g. limit number of generations or good enough score). Every iteration goes through "natural" processes such as selection, crossover, and mutation. The selection process includes the assessment of the individuals and then the selection of the strongest (according to the fitness function) individuals to reproduce and create the next generation, a combination of their genes (perpetuation of the best characteristics). The crossover and mutation bring diversity to the population that can potentially generate fitter individuals for the specific environment. These concepts are further explained in detail in a distinct publication by the same authors Fabian et al. [4]. Figure 3 illustrates the genetic algorithm process.

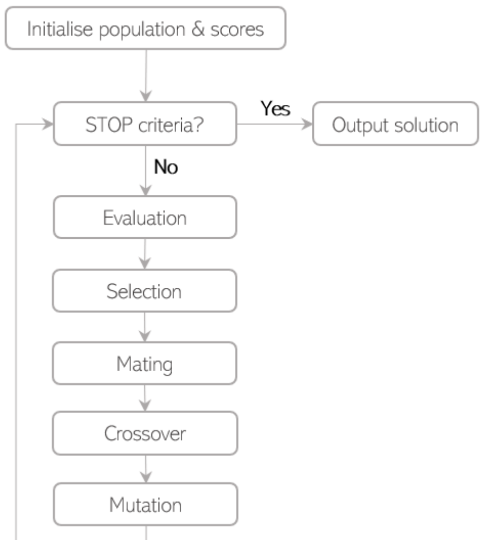


Fig. 3. Genetic Algorithm Process

III. Methodology

A. Experiment Set Up

This project makes use of two different data sources, Eurocontrol R&D [2] and OpenSky (2014, [16]). Eurocontrol data is used for the planned flight course, named in this project *filed data*, while OpenSky data was chosen for the recorded air traffic data, named *actual data*. There was a need to make use of both sources since Eurocontrol data does not have an available time step between data entries large enough for a good analysis of the recorded data, while OpenSky does. However, OpenSky only offers recorded data, not having any planned flight data stored.

The ATM controller simulator used is BlueSky (2016, [3]), an open-source software created with the goal to serve all researchers' needs without any restrictions, which can be downloaded and changed to fit each project's needs. The software is developed in *Python*, like the rest of all the research, and makes use of *PyQt* to interact with the user.

The main goal of this research is to compare the different time shifting methods, implemented in the scenarios run in BlueSky, and conclude which one is more suitable to eliminate the inherent human bias. The comparison is done method by method between a nominal scenario (a non time shifted scenario), defined in a section III-B (filed data vs. actual data), and the time shifting method. The parameters for this comparison rely on the same encounter characteristics distributions followed to identify the nominal scenario. The assessed encounter distributions, the same as the reference paper, focus on the horizontal and vertical distance between aircraft in an encounter, encounter relative headings (aircraft-to-aircraft) and encounter geometries (Level-Level, Level-Transitioning or Transitioning-Transitioning). The main time shifting method metric focuses on each method's effectiveness transforming and, secondly, maintaining the encounter distributions while approximating them to the nominal values, instead of focusing on the total simulation time as the reference paper. This research assigns a F value ($F \in [0, 1]$), a fitness score, obtained from the fitness function of the genetic algorithm, to classify each method.

Section III-B will describe the underlying principle to obtain the nominal scenario, while the time shifting methods implementation is portrayed more extensively in section III-C.

B. Filed vs. Actual Data

The nominal scenario is the reference for the different scenarios that apply the time shift. In other words, the goal is to obtain a distribution from the nominal scenario that does not take into consideration the pilot intervention or the ATCO clearances.

The nominal values, as mentioned, are obtained after a comparison between filed data and actual data scenarios. Nevertheless, on the one hand, the research approach from authors Paglione, Oaks, and Summerill (2003, [14]) chose the total nominal number of encounters as the reference to build the nominal scenario. It was obtained through hypothesis testing ("used to make a broad claim on the value of some population parameter or characteristic", 2003, [14]), deciding on some compromises and obtaining the total number of 23179 encounters. On the other hand, this research selected a three-hour scenario as the reference to build the nominal scenario and all nominal values are obtained as a function of that.

In addition, in the study where the method comes from, actual data was used to build the nominal scenario and obtain the different variables' distributions. However, the actual data should have all the unwanted human influence this research is looking to eliminate.

Therefore, a comparative study was run to assess the difference between the two three-hour scenarios, one from actual data and the other one from filed data.

C. Time shifting Methods Implementation

There was a necessity to adapt and tune several contributing variables for the different methods since this paper's conditions of simulation are not an exact replica of the already published research. Thus, the following segments will explore what the differences are within each method and why these were implemented.

It should be mentioned for future work on this subject that according to authors Paglione et al. (1999, [12]), mentioned in the time shift original paper (2003, [14]), a time shift of more than an hour can compromise the accuracy of the conflict probe trajectory modeler used. Therefore, even though the previous work was developed in a different conflict probe trajectory modeler, the value of one hour is assumed to be a fair approximation of the maximum allowed time shift.

1) Random Time Adjustment: The *random time adjustment* method generates randomly a vector of Δt elements, one element for each flight. The correspondent element of each flight is subtracted to every time data entry of the respective flight. Figure 4 illustrates this process and what happens to the trajectory.

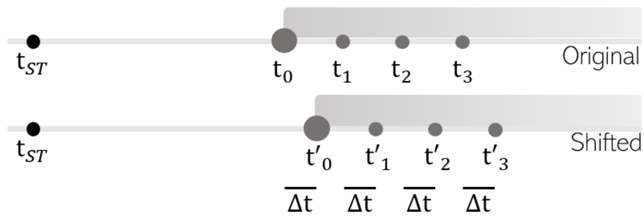


Fig. 4. Random time shift adjustment method for one trajectory

This method uses a random probability function following a gaussian distribution with a $\mu = 0$ and $\sigma = 900$.

2) Time Compression: The *time compression* method requires a time compression factor (t_c) common to all trajectories. This value, between 0 and 1, is multiplied by the difference between the first time entry (T_0) and the start time (T_{ST}), for every flight. The output T'_0, from equation 1, will then be crucial to compute the output Δt, from equation 2.

$$T'_0 - T_{ST} = (T_0 - T_{ST}) \times t_c \quad (1)$$

$$T_0 - T'_0 = \Delta t \quad (2)$$

Then, the same way as the *random time adjustment* method, the Δt is subtracted to every time data entry of the correspondent flight data, through the same process pictured in Figure 4.

For this research, a value of $t_c = 0.8$ and $t_c = 0.9$ are considered, in resemblance to the original one ($t_c = 0.75$) but considering the starting and total simulation time, so flights are not time shifted by more than one hour.

3) Genetic Algorithm Implementation: The time shifting implementation making use of a GA is more complex when compared to the previous methods. For this specific problem, the gene is each flight's Δt and the individual is the combination of all the Δt implemented in one simulation.

From the reviewed literature, this last procedure was the most suitable method to generate conflicts. Thus, even though the developed research concluded the best method out of the three was the *genetic algorithm implementation*, the results will be reproduced with some experiment setup alterations.

There are some genetic algorithm python libraries but, in order to efficiently run the algorithm, the *thread* class and the multi-processing feature (*Popen()*) were used and a genetic algorithm was built from scratch, based on the code from *Machine Learning Mastery* [10]. The full genetic algorithm is available on Github, as part of a larger project [5].

The GA has some characteristic parameters and these define the probability of crossover, the probability of mutation, the number of individuals, and, in this case, the limit on the number of generations (stop criterion). The potential and investigated values are in Table I (2003, [13]).

TABLE I
GENETIC ALGORITHM KEY PARAMETERS

Population (Nº. of individuals)	Crossover (%)	Mutation (%)	Generations (Nº. of iterations)
8	75, 90	3, 8	20, 50

Nº. of genes	Genes values
1016	random $\mathcal{N}(0, 600)$

The general approach of a genetic algorithm was introduced, along with the values of the general parameters. Nevertheless, some more specifications should be mentioned. The crossover technique was a two-point crossover technique, illustrated in Figure 5, while the mutation process replaced the original gene with a random integer between -3600 and 3600 (maximum time shift of one hour) and an elitism feature was implemented (2003, [13]). Elitism is an optional feature of genetic algorithms that preserves the best individuals from one generation to the following, with no crossover or mutation applied, transforming a parent into a child directly. The feature is used when the goal is to achieve the best score and since it can happen before the stop criterion is met, those solutions are kept. In this algorithm, for every generation (iteration), the best two individuals are copied to the following generation.

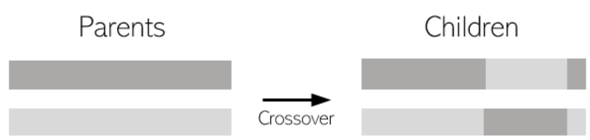


Fig. 5. Two point crossover technique

In addition, some focus on the evaluation process is crucial since this is the one that selects the fittest individuals. Each evaluation run considers all the variables in Table III and compares them to the set lower bound (LoB) and Upper Bound ($UppB$) which in the original paper were obtained by considering the LoB a third of the nominal number of encounters (23179 encounters) and the $UppB$ 10% more than that value, for each parameter. Nevertheless, in this research, the LoB and $UppB$ suffered some changes since the encounter number from the scenarios did not have the same dimension as in the original paper. Both LoB and $UppB$ are obtained from new nominal values (X), created as in Table II. All parameters are evaluated based on the distribution percentage, besides the *Number of encounters* variable.

The fitness function makes use of the values from Table II for each variable i and calculates the fitness value f ($f \in [0, 1]$), Equation 3. The independent variable $count_i$ maximizes the function for values within the bounds (LoB and $UppB$). The global fitness value is calculated simply as in Equation 4, where n is the total variable number.

TABLE II
VARIABLE BOUNDS FOR FITNESS FUNCTION IN GENETIC ALGORITHM

	Lower Bound (LoB)	**Upper Bound (UppB)**
Variable i	$X \times 0.9$	$X \times 1.1$

$$f(count_i) = \begin{cases} \frac{count_i}{LoB_i} & , count_i < LoB_i \\ 1 & , LoB_i \leq count_i \leq UppB_i \\ \frac{UppB_i}{count_i} & , count_i > UppB_i \end{cases} \tag{3}$$

$$\text{Global Fitness} = \frac{\sum_{i=1}^{n} f(count_i)}{n} \tag{4}$$

The calculations require information from the data log output provided by BlueSky. To obtain the information on LoS with the altered PZ parameters, the simulation is run with the Airborne Separation Assurance System (ASAS) switch ON, with the ASAS default settings adapted (PZ margins) and a look-ahead time of 0s, since an LoS detection and a conflict detection with a null look-ahead time are the same in the simulation conditions.

The data log is personalized with a resolution of one second (the same as flight data indications) and given variables. Some changes were implemented in the BlueSky python code so that the variables could include *simulation time*, *aircraft ID*, *distance* between aircraft, *aircraft performance flight phase*, *aircraft altitude*, and *aircraft heading*. Only the first data entry of a conflict is used and the same two aircraft can have a maximum of two distinct conflicts if these are spaced for more than 5 minutes, between the last alert of the first conflict and the first alert of the second one.

IV. RESULTS

This section encompasses all the results from this time shifting research, starting with a pre-inspection of the data type for the nominal scenario, in section IV-A. Then, in section IV-B, the core results are presented, from which some conclusions can be obtained.

The results focus on the total number of encounters for an original simulation of three hours (9 AM to 12 PM), on the 2^{nd} of September 2018. In a background analysis, different simulations dates were used to assess the veracity of the broad conclusions made for this specific case and different data sets present similar results. Thus, the selection of same duration data sets does not have a relevant impact on the results, but the total simulation time is an influencing factor even though its influence will not be explored.

For a better understanding of the results structure, the following tables include the analysis of different parameters organized in distinct sub-tables, each presenting independently the distribution of the total number of encounters. Hence, the encounters categorized in a specific bin in a sub-table can have a very different distribution in a different sub-table.

A. Filed vs. Actual Data

These results focus on the characteristics of the simulated air traffic scenarios where the main independent variable is the data type, filed data (Eurocontrol), or actual data (OpenSky).

TABLE III
NOMINAL SCENARIOS RESULTS FOR FILED AND ACTUAL DATA

	Filed data	**Actual data**
Number of encounters	5881 100 %	5767 100 %
Horizontal distance		
0nm to 5nm	1274 21.66 %	810 14.04 %
5nm to 10nm	1077 18.31 %	1225 21.24 %
10nm to 15nm	1047 17.80 %	1271 22.04 %
15nm to 20nm	1186 20.17 %	1234 21.40 %
20nm to 25nm	1287 22.06 %	1227 21.28 %
Vertical distance		
0ft to 500ft	3500 59.51 %	3215 55.75 %
500ft to 1000ft	377 6.41 %	252 4.37 %
1000ft to 2000ft	449 7.63 %	548 9.50 %
2000ft to 3000ft	555 9.44 %	495 8.58 %
3000ft to 4000ft	351 5.97 %	417 7.23 %
4000ft to 5000ft	649 11.04 %	840 14.57 %
Relative Heading		
0º to 30º	2390 40.64 %	1908 33.09 %
30º to 60º	790 13.43 %	1003 17.39 %
60º to 90º	861 14.64 %	939 16.28 %
90º to 120º	682 11.60 %	492 8.53 %
120º to 150º	474 8.06 %	483 8.38 %
150º to 180º	684 11.63 %	942 16.33 %
Encounter Geometry		
Level-Level	2469 41.98 %	1431 24.81%
Level-Transitioning	1369 23.29 %	1340 23.24 %
Transitioning - Transitioning	2043 22.41 %	2996 51.95 %

These simulations were reproduced with three different data sets as explained. In Table III, only one set of results is made available and it is possible to compare each encounter characteristic distribution for the two data types, knowing they are representative, with the absolute and relative values.

B. Time Shifting Methods

The results for each method are showcased, considering distinct parameters and the intrinsic random characteristic. The results in Table VII reveal each method's best performance, for the same initial scenario. The structure is similar to the previous results showing the absolute and relative values per category.

1) Random Time Adjustment:: Considering this method inherently has a random contribution, five different runs were made to assess the coherence of the results. These are showed in the Table IV below.

TABLE IV
F VALUES FOR THE RANDOM TIME ADJUSTMENT METHOD

	1st run	2nd run	3rd run	4th run	5th run
F value	0.905	0.903	0.899	0.909	0.900

2) Time Compression:: The results for the time compression method with the two different time compression factors are presented in Table V.

TABLE V
F VALUES FOR THE TIME COMPRESSION METHOD

	t_c = 0.8	t_c = 0.9
F value	0.893	0.888

3) Genetic Algorithm Implementation:: The values for the genetic algorithm implementation values are in Table VI, after running the algorithm with different parameters for number of iterations, crossover probability and mutation probability.

TABLE VI
F VALUES FOR GENETIC ALGORITHM IMPLEMENTATION

Number of iterations	Crossover probability [%]	Mutation probability [%]	F value
20	75	3	0.9243
		8	0.9293
	90	3	0.9287
		8	0.9296
50	75	3	0.9402
		8	0.9338
	90	3	0.9455
		8	0.9367

The evolution for the genetic algorithm implementation is illustrated below in Figure 6, for 50 iterations, 90% crossover probability and 8% mutation probability.

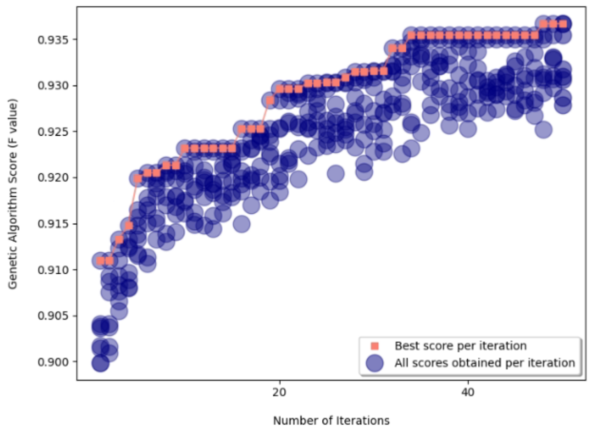


Fig. 6. Genetic evolution for 90% crossover and 8% mutation probabilities

4) Assembly of the different time shifting methods: The best runs for each method are in Table VII.

TABLE VII
TIME SHIFTED SCENARIOS WITH DIFFERENT TECHNIQUES FOR ACTUAL DATA

	Time Compression	**Random Time Adjustment**	**Genetic Algorithm**
Number of encounters	7063	5405	5526
Horizontal Distance			
0nm to 5nm	1068 15.12 %	934 17.28 %	1068 19.33 %
5nm to 10nm	1500 21.24 %	1095 20.26 %	1102 19.94 %
10nm to 15nm	1532 21.69 %	1146 21.20 %	1088 19.69 %
15nm to 20nm	1480 20.95 %	1123 20.78 %	1177 21.30 %
20nm to 25nm	1483 21.00%	1107 20.48 %	1091 19.74 %
Vertical Distance			
0ft to 500ft	3893 55.12 %	2965 54.86%	3101 56.12 %
500ft to 1000ft	342 4.84 %	299 5.53 %	317 5.74 %
1000ft to 2000ft	675 9.56 %	540 9.99 %	521 9.43 %
2000ft to 3000ft	638 9.03 %	472 8.73 %	497 8.99 %
3000ft to 4000ft	498 7.05 %	391 7.24 %	368 6.66 %
4000ft to 5000ft	1017 14.40 %	738 13.65 %	722 13.06 %
Relative Heading			
0º to 30º	2387 33.80 %	1850 34.23 %	1922 34.78 %
30º to 60º	1233 17.46 %	950 17.58 %	964 17.44 %
60º to 90º	1100 15.57 %	867 16.04 %	826 14.95 %
90º to 120º	615 8.71 %	477 8.82 %	550 9.95 %
120º to 150º	578 8.18 %	426 7.88 %	412 7.46 %
150º to 180º	1150 16.28 %	835 15.45%	852 15.42 %
Encounter Geometry			
Level-Level	1763 24.96 %	1331 24.63 %	1327 24.02 %
Level-Transitioning	1606 22.74 %	1245 23.03 %	1249 22.60 %
Transitioning - Transitioning	3694 52.30 %	2829 52.34 %	2950 53.38 %

V. DISCUSSION

A. Filed vs. Actual Data

From Table III, the first comparison is over the total number of aircraft encounters since filed data has approximately three and a half times more encounters than the actual data. The total encounter number difference follows the expected qualitative pattern, considering the total aircraft number is not the same (actual data simulation made use of two-thirds of the filed data simulation aircraft). There is no direct theoretical correlation between the number of aircraft and the consequent encounters, therefore it is not possible to assess whether the absolute number of encounters, proportionally, is correct.

Nevertheless, it is possible to compare the relative values of each bin and, taking into consideration the first observation, these are the values where the focus should be. The overall analysis of the parameters raises some attention to the first 0-5NM *Horizontal Distance* bin and for the *Encounter Geometry* bins (Level-Level and Transitioning - Transitioning).

Firstly, from all the horizontal distance bins, the first one is the one that presents distinct distributions with a percentage of 21.66% for filed data and 14.04% for actual data. This comparison was one of the main goals of the pre-investigation since the suspicion relied on whether the actual data characteristics were fit to be considered the nominal scenario characteristics (with no human bias). The conclusions deduced from the data are that the actual data has a lower encounter count with a distance of 0-5NM, the common borderline distance at which an encounter is considered a Loss of Separation. In other words, due to conflict prediction and resolution, it accounts for pilot intervention and ATC ground controllers' indications. Nevertheless, the existence of encounters with less than 5NM distance horizontally can be explained by the fact that those encounters can have a vertical distance higher than 1000ft, not being a real safety issue, or by the fact that they didn't happen in an area with a minimum horizontal safe distance of 5NM, such as when surveillance systems' capabilities allow it, where the minimum horizontal safe distance can be 3NM.

Secondly, the encounter geometry was not a decisive factor but it's possible to see also a discrepancy in the values comparing the two data type columns. The major difference is in the Level-Level and the Transitioning-Transitioning geometries. The contrasting data can have different origins but the ones that are considered most likely are route alterations (more direct paths), possibly eliminating some Level-Level encounters, and airport logistic delays, possibly generating an intricate operations environment for the air traffic controllers.

Finally, from the different observations but focusing on the 0-5NM horizontal distance bin, the conclusion leads to building a nominal scenario with the filed data where there is no influence of pilot intervention or ATC clearance commands.

B. Time Shifting Methods

All time shifting methods transform as expected the actual data scenarios into scenarios more similar to the filed data ones, demonstrating higher F values than the actual data F value ($F = 0.883$).

The *random time adjustment* method was run five times to assess how its non-deterministic character influenced the results. The five F values are close and coherent, but verify the impact of the method's random nature.

The *time compression* method is the only one with deterministic characteristics and a direct limitation in its parameters since the time compression factor has to consider no time shift should be higher than an hour. The results showed the same closeness and coherence as the previous method, even though with an overall lower F value trend. From the simulations obtained, the first method is more effective than this one, but it is important to highlight its random nature since it can generate in some cases scenarios with lower performance than the deterministic *time compression*s method.

The *genetic algorithm implementation* showed the same coherence within the method as the previous ones mentioned. The F values are all distinctively higher than the two previous methods, something expected considering the nature of the algorithm and how it was built. Figure 6 displays the evolution of one run of the genetic algorithm, illustrating the normal algorithm's tendency to have a steeper improvement for the initial generations (crossover contribution mainly) and convergent evolution afterward. The convergent character of this algorithm happens due to the loss of initial diversity, meaning the crossover's influence starts to decrease, while the mutation becomes the main diversity contributor.

The comparison between the three methods' performance in detail is crucial to understand if the F value obtained explores the transformation goals as it was desired.

The comparison of each method's distribution is done always considering the nominal values, mainly looking at the criteria used to pick the nominal values between filed data and actual data. The focus is on the 0-5NM bin for *Horizontal Distance* and the *Encounter Geometry* options. The 0-5NM bin shows clear evolution and classification similar to the F values with the genetic algorithm method reaching 19.33%, thus closer to the nominal values (21.66%) than the initial ones (14.04%). Concerning the *Encounter Geometry* options, the evolution was low and the values resemble considerably more the actual data than the filed data. This progress was not better in any of the three methods, and as a consequence, this contribution was set aside for performance comparison. A criterion not mentioned before due to the similarity between filed data and actual data is the total number of encounters in the scenario. This variable does not raise any concerns for the *random time adjustment* and the *genetic algorithm implementation*, but it stands out for the *time compression* method with a total encounter number of 7063 against the 5881 in the nominal values. Overall, the F values attribution and the method's rank are a match to the variables' distributions goals.

VI. Conclusion

The purpose of this paper is to assess and compare different time shifting techniques to eliminate the human bias existent in recorded air flight data, due to conflict solving actions. The methods considered were *time compression*, *random time adjustment*, and *genetic algorithm implementation*. A previous study on this topic was used to keep some parallelism but some research choices were redesigned and the three methods were built from scratch. The new design choices included defining the nominal values and new assessment criteria.

The nominal values, by principle, should not include the mentioned human bias present in recorded air flight data, since the goal is to eliminate this effect. After the encounters' characteristics comparison between actual data and filed data, the filed data was the logical choice for the nominal data. The criteria to assess the best time shifting method was defined as the best method to transform the initial scenario, concerning the encounter characteristics' distributions in percentage, into a scenario with filed data characteristics.

The three methods revealed some success in transforming actual data scenarios into scenarios with filed data characteristics. Nevertheless, the *genetic algorithm implementation* was the most effective method according to the criteria provided, followed by the *random time adjustment* and, then, *time compression*. The last two methods are considerably more computationally efficient than the first one.

This research focused on contributing to the Air Traffic Management community by working with recorded air traffic data and realistic simulation settings. To complement this research, new genetic algorithm fitness functions could be considered (integration of weighted variables or new encounter characteristics) and more complex methods could be created by conjugating the already assessed ones (e.g. *time compression* with *random time adjustment* or *genetic algorithm implementation*).

References

[1] Sameer Alam et al. "Evolving air traffic scenarios for the evaluation of conflict detection models". In: *6th EUROCONTROL Innovative Research Workshop and Exhibition: Disseminating ATM Innovative Research* (2007), pp. 237–245.

[2] *Archive data for R&D - Eurocontrol [online]*. https://ext.eurocontrol.int/prisme_data_provision_hmi/. Accessed: 2021-03.

[3] Joost Ellerbroek and Jacco M Hoekstra. "BlueSky ATC Simulator Project: an Open Data and Open Source Approach Three-Dimensional Airborne Separation Assistance Displays View project BlueSky-Open source ATM simulator View project BlueSky ATC Simulator Project: an Open Data and Open Source Approach". In: *seventh International Conference for Research on Air Transport (ICRAT)* (2016).

[4] Andrew J Fabian et al. "Design and Performance of an Improved Genetic Algorithm Implementation for Time-Shifted Air Traffic Scenario Generation". In: (2016).

[5] *Github Repository [online]*. https://github.com/lunajuliao/CDperformance. Accessed: 2022-05.

[6] Sehchang Hah, Ben Willems, and Randy Phillips. "The effect of air traffic increase on controller workload". In: *Proceedings of the Human Factors and Ergonomics Society [e-journal]* (October, 2006), pp. 50–54.

[7] ICAO. *Doc 4444 - Air Traffic Management - Procedures for Air Navigation Services*. 2016, pp. 5.1–5.81.

[8] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, 1992.

[9] Chan Li Long, Yash Guleria, and Sameer Alam. "Air passenger forecasting using Neural Granger causal Google trend queries". In: *Journal of Air Transport Management [e-journal]* 95 (2021), p. 102083.

[10] *Machine Learning Mastery [online]*. https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/. Accessed: 2021-05.

[11] William P. Niedringhaus. "Solution Complexity Metrics". In: *1998 Guidance, Navigation, and Control Conference and Exhibit* (1998), pp. 30–51.

[12] Mike M Paglione et al. "User request evaluation tool daily use time shifting trajectory prediction accuracy degradation study". In: *Time* (1999).

[13] Mike M. Paglione, Robert D. Oaks, and Karl D. Bilimoria. "Methodology for generating conflict scenarios by time shifting recorded traffic data". In: *AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Forum* (2003).

[14] Mike M. Paglione, Robert D. Oaks, and J. Scott Summerill. "Time shifting air traffic data for quantitative evaluation of a conflict probe". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit* (2003).

[15] Russell A. Paielli. "Empirical Test of Conflict Probability Estimation". In: *2nd USA/Europe Air Traffic Management R&D Seminar (ATM-98), Orlando, Florida* (1998).

[16] Matthias Schäfer et al. "Bringing up OpenSky: A large-scale ADS-B sensor network for research". In: *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. 2014, pp. 83–94.

3

State-based Conflict Detection Performance as a Function of Look-ahead Time

State-based Conflict Detection Performance as a Function of Look-ahead Time

Luna Julião

Abstract—**Airspace's increasing demand is a current concern without a solution. Different research projects aim to expand its available capacity by improving software performance, for example, concerning trajectory prediction and conflict detection methods. This research contributes to this goal by investigating the effect of different look-ahead time values on a state-based conflict detection method's performance. A parallel analysis is made concerning the traffic density and meteorological conditions' influence.**

The simulations use actual air traffic data, obtained from the OpenSky database. Corrective data processing is implemented to minimize the noise due to data resolution issues, and time-shifting techniques are analyzed and implemented to counteract the human bias natural in real recorded data. The chosen air traffic simulator is the open-source BlueSky Simulator, which integrates the state-based method analyzed.

The data is selected considering the traffic density (Eurocontrol database) and the meteorological conditions (ERA5 data from Climate Copernicus) since *Light*, *Medium* and *High* bins are created. The traffic density bins generation makes use of *k-clustering*, while the meteorological conditions bins go through a more complex process to identify atmospheric cold fronts.

The performance is obtained for different classification approaches, showing the impact more flexible metrics have on the results. For flexible metrics, the performance of the state-based conflict detection method is higher than for stricter metrics. For the first one mentioned, values higher than 120s look-ahead time are not fruitful, while, for the second one, all look-ahead times are not effective in state-based conflict detection. An analysis focusing on the flight phase showed the performance is better for the cruise phase, raising the effective look-ahead times to 300s and 180s for each approach, respectively.

Concerning the secondary independent variables, firstly, a higher traffic density environment translates to a lower conflict detection performance. Secondly, the meteorological conditions bins' difference is not enough to withdraw conclusions, even though it follows a similar trend to the traffic density values.

Index Terms—**Conflict Detection, State-based Method, Look-ahead Time, Performance Assessment, OpenSky, Time Shift Scenario, Genetic Algorithm, BlueSky, Meteorological Conditions, Traffic Density**

I. Introduction

IN the aviation sector, a growing concern focuses on the balance between airspace capacity and demand (2006, [12]). The airspace is limited, but, in theory, its capacity can still expand to satisfy the increasing demand. To achieve this, different researchers focused on this goal through distinct techniques, some working on the Air Traffic Control (ATC) ground controllers' workload and perception ([12]), while others on the technical improvements the current conflict detection software concept could include (2000, [4] and 2002, [3]) or, in some cases, alternatives to the current system (2001, [14] and 2005, [21]).

This project focuses on obtaining ATM performance results, looking into specific conflict detection parameters. The initial and main goal is to obtain representative and realistic results, thus, contributing to the air traffic research community. The design choices for this research were based on previous work on this topic while attempting to remain as close as possible to the real data.

This research works with raw ADS-B data, obtained from OpenSky (2014, [24]) with a data resolution of 1 second, in contrast to modeled data (2011, [17] and 2020, [29]). Other analyses use radar data (2009, [26]), nevertheless, by applying parametric modeling or with a considerably higher resolution.

Recorded air traffic data, if used for aircraft conflict research, needs an extra processing step to minimize the human intervening actions aimed at conflict prevention (e.g. air traffic controllers' indications or pilots' conflict resolution actions). Time shifting is one possible approach introduced by authors Alam et al. (2007, [1]), who mentioned other strategies (1998, [18], [22]). This approach was developed by Alam et al. who employed a genetic algorithm, but Paglione, Oaks, and Summerill (2003, [20]) also looked into the time shifting technique and developed it by coming up with three different methods to generate conflicts with a real-life representative distribution. These three methods will be studied in this research: *random time adjustment*, *time compression*, and *implementation of a genetic algorithm* to obtain fitter time shifts.

To obtain a comprehensive safety analysis, as noted by authors Sunil et al. (2018, [25]), it is essential to consider the three different flight phases: climb, cruise and descent. Sunil et al. concluded that climb and descent contributed with the majority of the conflicts for unstructured layered airspace design. Similarly, authors Lauderdale, Cone, and Bowe concluded descent-speed errors and top-of-descent errors were the main conflict cause when compared to other causes such as weight and wind errors. These conclusions will be considered when looking at the results since the research on hands encompasses all flight phases.

The conclusions are obtained by comparing the performance of the conflict detection software with different values for the look-ahead time parameter, considering a set of thirty-six scenarios. These scenarios are specifically selected so that diverse characteristics are assessed, namely the traffic density and meteorological conditions.

This process relies on the open-source BlueSky ATC Simulator (2016, [5]).

II. Problem Statement

A. Research objective

The goal of this research is to contribute to the comprehension and evaluation of the Air Traffic Control tools' accuracy by conducting a performance assessment, as a function of look-ahead time, of a state-based conflict detection software integrated into the open-source ATM BlueSky Simulator.

B. Background theory

This research makes use of fast-time air traffic simulation to achieve its objective. Air traffic simulators, focusing on the ones used by ATC ground controllers, incorporate conflict detection and resolution features. A detailed look at the conflict detection functionality and the basic concepts inherent to this topic is presented. Then, a summary of the *genetic algorithm* is presented considering its importance in the research unfolding.

Firstly, a conflict detection software relies on flight data (state-based method) and, possibly, on flight plan data (intent-based method) to extrapolate each aircraft's future trajectory. Then, the calculations to identify conflicts are processed. In this research, only the state-based method will be assessed, available in the ATM BlueSky Simulator [5]. Secondly, the basic concepts that require comprehension are loss of separation (LoS), aircraft conflict, and look-ahead time. The concept *loss of separation* happens when an aircraft crosses another aircraft's Protected Zone (PZ). As established by ICAO, when using surveillance systems [15], an aircraft's PZ, in most cases, is 5 nautical miles (NM) horizontally and 1000ft vertically (Figure 1). The concept *aircraft conflict* concerns the potential future LoS detection, according to a prediction in time, the look-ahead time. This concept is an arbitrary parameter, required in conflict detection, that sets the predicted trajectory length in time. Thus, if the look-ahead time is 2 minutes, the software predicts each aircraft's trajectory 120 seconds into the future.

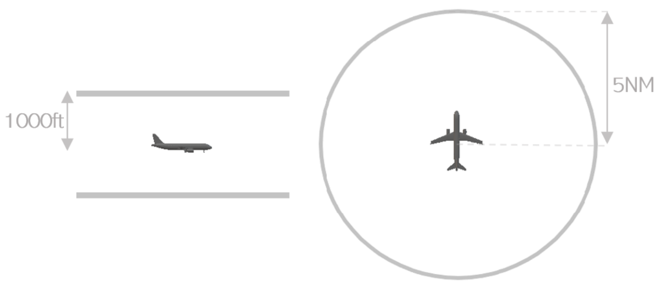


Fig. 1. Protected Zone representation

The genetic theory applied to computational algorithms was first introduced by Holland (1992, [16]). Genetic algorithms (GA) get their fundamental principles from biological phenomena and Charles Darwin's concept of natural selection. Natural selection is a natural process and results in the selection, within populations, of the fittest individuals to a specific environment. The migration into computational algorithms happens through a fitness function where the input elements with a higher fitness function output are selected. The basic GA key concepts are clarified below, along with Figure 2 which exemplifies the concepts according to how they are used in this work.

- Individual / Chromosome: An individual or a chromosome, for the specific case both can be considered the same, are the elements of a population. In GA, each one is a potential solution to the problem in question.
- Population: A group of individuals that will reproduce the next generation and from where the fittest elements are selected.
- Gene: A gene is the element of the individual/chromosome, thus always correlated with a specific parameter of the problem's solution.

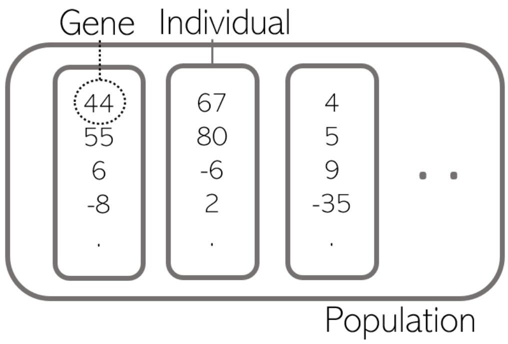


Fig. 2. Genetic Algorithm basic elements

Genetic Algorithms are stochastic algorithms with no guarantee of a good solution, iterated until a stop criterion is fulfilled (e.g. limit number of generations or good enough score). Every iteration goes through "natural" processes such as selection, crossover, and mutation. The selection process includes the assessment of the fittest (according to the fitness function) individuals to reproduce and create the next generation, a combination of their own genes (perpetuation of the best characteristics). The crossover and mutation exist to bring diversity to the population and potentially generate fitter individuals to the required environment. These concepts are further explained in detail in a separate publication, mentioned in the introduction, by the same authors Fabian et al. [9].

III. Methodology

This research works in parallel with distinct topics that are brought together for the processing and framing of the results that answer the set research objective. This section includes a detailed explanation of how the addressed topics crucial to this research correlate and culminate in the final results. Figure 3 illustrates how the different subjects are organized. The Flight Data (*C*) and Scenario Classification (*D*) encompass very distinct processes, hence, the creation of the sections *C1*, *C2*, *D1* and *D2*.

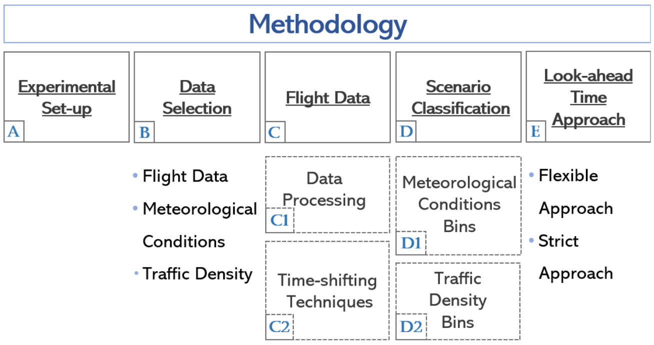


Fig. 3. Methodology Structure

The divisions in Figure 3 are the different steps pursued along with this project. Topic A focuses on the research plan and design choices chosen for this research objective, while topic B covers the first steps into the work, by searching and obtaining the data so that topics C, D, and E could be put into action.

The results for topics C and D are presented in section IV (*Preliminary Results & Discussion*), since both concern parallel background work necessary to frame and obtain the final results. These are the Look-ahead Time Approach's results, topic E, which are included in the *Results* section and discussed in the *Discussion* of the present work, due to their relevance and importance to the research objective. This organization is clarified in Fig. 4.

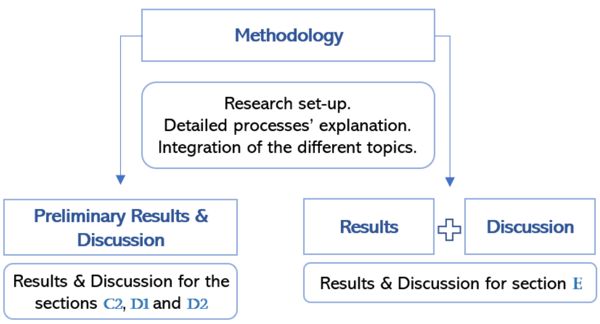


Fig. 4. Research Structure

A. Experimental set up

The goal of this research is to assess the performance of conflict detection software while varying one main independent variable, the look-ahead time. The main variable is correlated with the ATM simulation software BlueSky, more precisely with the state-based conflict detection method made available. The secondary independent variables considered are the aircraft traffic density and the meteorological conditions (atmospheric cold fronts generation).

This research uses a variation of software and algorithms due to the different preliminary steps necessary to obtain the final results. On the one hand, air traffic data needs to be processed and transformed, as explained below, while, on the other hand, the meteorological conditions and traffic density variables also have to be correctly considered and included in the ensemble of the research objective.

Firstly, the Air Traffic Controller simulator used is BlueSky [5], an open-source software created to assist all researchers' needs without any restrictions. The most used commands in BlueSky here are *CRE*, *MOVE*, *ALT* and *DEL*, besides some initial parameters to define the simulation characteristics and the data logging features. The input simulation characteristics are 0.5 seconds simulation time step, fast-time simulation indication for the whole duration of the scenario, the look-ahead time, and the Airborne Separation Assurance System (ASAS) switch ON. The data log is personalized with a resolution of one second (the same as flight data indications) and given variables. Some changes were implemented in the BlueSky python code so that the variables could include *simulation time*, *aircraft ID*, *distance* between aircraft, *time to loss of separation*, *aircraft performance flight phase*, *aircraft altitude*, and *aircraft heading*. The BlueSky software is developed in *Python* as all the research, which is made available on Github [11].

Secondly, as explained in section III-C, the initial focus is on obtaining and processing the air traffic recorded data through several steps, by transforming the data with different techniques (outlier elimination, data interpolation, low-pass filtering, and moving average). Also, in order to eliminate the existing bias due to human conflict solving and avoiding actions in recorded data, time shifting techniques are assessed, starting by using the framework and methods (*random time adjustment*, *time compression* and *genetic algorithm implementation*) based on Paglione, Oaks, and Summerill [20].

Finally, before addressing this research's final results, the last step is integrating the meteorological conditions and traffic density variables, which is achieved by constructing bins for the scenarios selection (tackled in section III-D). Each flight data subset (6-hour simulation) is picked from a large database fragmented into 6-hour segments. These fragments are put into *Light*, *Medium* or *High* categories (three bins) for each variable. The necessary preliminary step is defining the bins. For example, the meteorological conditions bins are considered by analyzing the existence of atmospheric cold fronts due to the impact they have on potential aircraft reroutes since for some scenarios flying through a cold front can be extremely dangerous. The *Light* bin includes all flight data fragments that are considered as comfortable weather to fly, while the *Medium* bin includes situations with some weather perturbations, and, finally, the *High* bin includes fragments with more severe weather events. A similar scale of *Light*, *Medium*, and *High* is used for the traffic density bins.

The final results focus on the conflict detection algorithm's performance while varying the look-ahead time, focusing on the considered values: 2min, 3min, 5min, 7min, 10min, and 15min. The performance evaluation relies on the confusion matrix's parameters by considering *true positives* (TP), *false positives* (FP) and *false negatives* (FN). The metrics for these variables affect considerably the final results, hence, the criteria to obtain TP, FP and FN are explained in detail for both the *flexible approach* and *strict approach*. The performance parameters are calculated with information obtained from the data logging file, created while the BlueSky simulation is running.

B. Data selection

1) Flight data: This project makes use of two different data sources, Eurocontrol R&D [2] and OpenSky [24]. Eurocontrol data is used for the planned flight course, named in this project *filed data*, while OpenSky data was the chosen source for the recorded air traffic data, named *actual data*. It was necessary to use both sources since Eurocontrol data does not have an available time step between data entries large enough for a good analysis of the actual data, while OpenSky does. However, OpenSky only has available recorded data, not having any planned flight data.

The specific date and time data details can be found in Table X. The geographical area was limited due to data sets' dimensions and the total area goes from 50°47'52"N, 01°24'29"E to 53°53'13"N, 7°25'3"E.

2) Meteorological data: The meteorological data had different possible data sources, considering the characteristics required for the analysis. The data used, in the end, is ERA5 hourly data obtained from Copernicus Climate Change Service (C3S) Climate Data Store [13]. This data is organized in a lat-long grid with a resolution of 0.25°x0.25°, obtained by combining model data with weather reports from all over the world in a data set using the laws of physics [6].

The information obtained covered a period from 07:00 AM to 08:00 PM every day from January to December, from 2017 to 2019. The selected geographical area is coherent with the flight data information (from 48°00'00"N, 01°00'00"E to 55°00'00"N, 08°00'00"E). The meteorological parameters collected, at pressure level 900hPa, were *relative vorticity* and *temperature*.

3) Traffic density data: The traffic density data was obtained from the Eurocontrol Aviation Intelligence Portal [8], under the section *En-route IFR flights and ATFM delays*. These files focus on flight delays but also incorporate the total number of aircraft within a specific airspace [7]. The selected airspace data corresponded to the *Belgium* and *Netherlands* Flight Information Region (FIR), matching approximately the flight data geographical limits.

The yearly datasets were obtained for 2017, 2018, and 2019.

C. Flight data processing

1) Data improvements & assumptions: The data obtained from OpenSky when processed revealed some unrealistic inconsistencies, for example, in the horizontal and vertical axis, namely the horizontal and vertical velocity being non-coherent with the consecutive horizontal and vertical positions, respectively. In order to redress the issues, crucial to the simulation and conflict detection, some amend actions were implemented.

- Action 1 - Duplicate information for the same horizontal coordinate points, for a flying aircraft, was deleted.
  In the data obtained, consecutive data entries with different timestamps present the same latitude and longitude coordinates for a flying aircraft, which can not happen in real life. Action 1 keeps the first data entry, resulting in timestamp jumps of, usually, two seconds to three seconds, but in some cases going up to twenty seconds.
- Action 2 - An interpolation is made to obtain the flight information per second.
  The data obtained has, in some cases, breaches of information in time, incremented with Action 1. Hence, an interpolation (one-second resolution) helps to bridge this situation.
- Action 3 - The vertical rate data had outliers eliminated, and it was then replaced with its low-pass filtered moving average (20 data entries total) version.
  The outlier elimination was executed before Action 2 to minimize the risk of accepting an outlier as an acceptable data entry. The low-pass filtering made use of the *scipy.signal* library, specifically, the Butterworth filter (0.1Hz cut-off frequency, 1Hz sampling frequency, and order 2).
- Action 4 - The altitude data was replaced with new values, calculated with the first altitude data entry and, for each time step, with the vertical rate obtained in Action 3.
  This substitution transformed vertical trajectories into smoother and coherent trajectories.
- Action 5 - Horizontal velocity was processed to eliminate outliers. The bearing data is improved through a low-pass filter process. In addition, new latitude and longitude coordinates were obtained from the initial conditions, the obtained velocity, and the heading.
  The horizontal axis data, lat-long coordinates, did not match the velocity given, made clear by the calculation of the instantaneous velocity for each time step. Thus, the necessity for coherence led to the recalculation of the horizontal trajectory with the velocity parameters.
- Action 6 - All aircraft considered *Light* under the Wake Turbulence Category (WTC) are deleted.
  This research work aims at aircraft considered non-military and non-general aviation. Therefore, the flag for general aviation and air vehicles that are to be excluded was determined as all the vehicles from the *Light* WTC (assessed with WTC database from UK Civil Aviation Authority [28], complemented with Flightradar24 database [10]).
- Action 7 - Any individual flight with less than 30 seconds of recorded data is disregarded.
  Flights with less than 30 seconds (data entries) are disregarded for two motives. Firstly, for our goal, short flights don't have relevance considering the smallest look-ahead time is 2 minutes. Secondly, some flight IDs come with an error for a short time interval, under 30 seconds, and since this information can not be matched to the correspondent flight (e.g. main flight's callsign being *RJA602* and the incorrect one *BY G F*), it can not be considered.
- Action 8 - All flights that are not recognized by the Fligh-

tRadar24 database [10] or are not included in the aircraft database (obtained in OpenSky [19]) are disregarded.
The BlueSky Simulator requires the aircraft model of aircraft to better simulate and predict future trajectories. Therefore, from the data selected, all flights were analyzed and information was included if it was available in the mentioned databases.

2) Time shifting methods: Recorded flight data requires the use of techniques, as mentioned, to eliminate the human controller bias since it is being used to assess conflict detection performance, thus, before the human takes action. For this, the initial study follows the framework used by authors Paglione, Oaks, and Summerill [20], looking at three time shifting methods: *random time adjustment*, *time compression*, and the *genetic algorithm implementation*.

The *random time adjustment* method generates randomly a vector of Δt elements, one element for each aircraft. The correspondent element of each aircraft is subtracted to every time entry of each aircraft. Figure 5 illustrates this process and what happens to the trajectory.

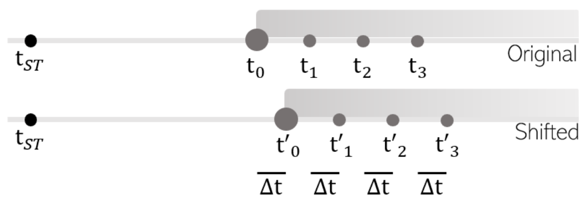


Fig. 5. Time shift adjustment method for one trajectory

The *time compression* method requires a time compression factor (t_c) common to all trajectories. Then, this value, between 0 and 1, is multiplied by the difference between the first time entry (t_0) and the start time (t_{ST}), for every flight. The output t'_0, from equation 1, will then be crucial to compute the output Δt for each flight, from equation 2.

$$t'_0 - t_{ST} = (t_0 - t_{ST}) \times t_c \tag{1}$$

$$t_0 - t'_0 = \Delta t \tag{2}$$

Then, in the same way as *random time adjustment*, the Δt is subtracted to every time entry of the correspondent flight data, through the same process pictured in Figure 5.

The time shifting implementation making use of a genetic algorithm is more complex when compared to the previous methods. For this specific problem, the gene is each aircraft's Δt and the individual is the combination of all the Δt implemented in one scenario (one simulation run).

This research takes a step in a different direction, compared with Paglione, Oaks, and Summerill's research [20], by selecting the filed data as a reference frame for the fitness function. The reference values are obtained from the Closest Point of Approach (CPA) distribution, for every aircraft pair interaction up to 25NM and 5000ft. In addition, the selected variables to be evaluated are the same but used as a percentage since the absolute value is considerably different from case to case. The *Number of Encounters* parameter is the only absolute value and is adjusted for every scenario. The other variables from *Horizontal Distance*, *Vertical Distance*, *Relative Bearing* and *Encounter Geometry* are evaluated in percentage, using the reference values (X) of the simulation with filed data within an interval (Table I and Table II).

The fitness function makes use of the values from Table I for each variable i and calculates the fitness value f ($f \in [0, 1]$), Equation 3. The independent variable $count_i$ maximizes the function for values within the bounds (LoB and $UppB$). The global fitness value is calculated simply as in Equation 4, where n is the total variable number.

The results' calculation requires alterations, before the simulation, in the BlueSky simulation settings, and information from the data log output provided by BlueSky after the simulation. On the one hand, before the simulation, the PZ margin settings should be changed to 25NM horizontally and 5000ft vertically, and the look-ahead time set to 0s. On the other hand, the variables used from the data log are all the ones mentioned before, apart from the *time to LoS*. In addition, only the minimum *distance* data entry is used for each encounter (two aircraft have a maximum of two encounters if these are spaced for more than 5 minutes).

TABLE I
VARIABLE BOUNDS FOR FITNESS FUNCTION IN GENETIC ALGORITHM

	Lower Bound (LoB)	**Upper Bound (UppB)**
Variable i	$X \times 0.9$	$X \times 1.1$

TABLE II
NOMINAL SCENARIO RESULTS FROM FILED DATA

Horizontal distance	**Filed data**
0nm to 5nm	21.66 %
5nm to 10nm	18.31 %
10nm to 15nm	17.80 %
15nm to 20nm	20.17 %
20nm to 25nm	22.06 %

Vertical distance	**Filed data**
0ft to 500ft	59.51 %
500ft to 1000ft	6.41 %
1000ft to 2000ft	7.63 %
2000ft to 3000ft	9.44 %
3000ft to 4000ft	5.97 %
4000ft to 5000ft	11.04 %

Relative Heading	**Filed data**
0º to 30º	40.64 %
30º to 60º	13.43 %
60º to 90º	14.64 %
90º to 120º	11.60 %
120º to 150º	8.06 %
150º to 180º	11.63 %

Encounter Geometry	**Filed data**
Level-Level	41.98 %
Level-Transitioning	23.29 %
Transitioning - Transitioning	22.41 %

$$f(count_i) = \begin{cases} \frac{count_i}{LoB_i} & , count_i < LoB_i \\ 1 & , LoB_i \leq count_i \leq UppB_i \\ \frac{UppB_i}{count_i} & , count_i > UppB_i \end{cases} \quad (3)$$

$$\text{Global Fitness} = \frac{\sum_{i=1}^{n} f(count_i)}{n} \quad (4)$$

D. *Scenarios classification*

The scenarios can be classified, as mentioned, in three bins: *Light*, *Medium* and *High*, for each one of the variables meteorological conditions and traffic density. The following sections elaborate on the process to define and obtain these bins.

1) Meteorological Bins: The elected indicator to categorize six-hour data into Light (L), Medium (M), and High (H) meteorological bins was the existence of atmospheric fronts, namely cold fronts. The research on *A simple diagnostic for the detection of atmospheric fronts* [23], from authors Parfitt, Czaja, and Seo, guided the steps of the ERA5 data transformation into the identification of the atmospheric front.

Firstly, the process relied on computing a variable f, Equation 5, with the relative vorticity (ζ_p) and the temperature gradient ($|\nabla(T_p)|$), calculated from the given ERA5 temperature data. Secondly, the variable f is normalized with $|\nabla T|_0 = 0.0045 K/km$ and c (Coriolis parameter at the given latitude) to compute the final parameter F, Equation 6. The F parameter is calculated for each grid cell (0.25°x0.25°).

$$f = \zeta_p \, |\nabla(T_p)| \quad (5) \qquad F = \frac{f}{c\,|\nabla T|_0} \quad (6)$$

In the research introduced, the identification metric relied on $F > 1$ corresponding to a cold front, while any other value below one would result in a uneventful scenario. However, the F values obtained were not coherent with this metric, since the vast majority of the days would have F values higher than 1 for some coordinate points. Hence, a new metric was implemented.

The new identification metric considered the F values and initially went through a k-means clustering method to find possible reference points for the three bins. These reference points were later tuned by selecting random days and matching the temperature gradient and precipitation per city (obtained from a different weather report data source [27]) with the F values, obtaining an L, M, and H categories for each grid cell. Then, these categories were used for the final metric, which established the balance between the three categories required for the *Light*, *Medium*, and *High* final bins. These metrics are further explained in the Github repository [11] under *Scenario Classification*, and a quantitative summary is presented in section IV-B.

2) Traffic density bins: The traffic density data was handled bringing together the three-yearly files. The data was, then, processed by looking at the Flight Information Region (FIR), selecting the Netherlands and Belgium regions.

The following step makes the assumption the daily flight number, in the data, is the sum of an even distribution throughout the day and that the aircraft elimination ratio, from the database, of general aviation is similar every day.

The days were categorized into three bins (*Light*, *Medium*, and *High*) by summing the total daily flight number (variable *FLT_ERT_1*) of both FIR and using the k-means clustering method, available in the *KMeans* library in Python.

E. *Look-ahead Time Approach*

Look-ahead time is a parameter usually used in trajectory prediction for conflict detection and it concerns the extension of the prediction. A look-ahead time of one minute, for a state-based method, translates to a linear prediction of the trajectory, based on the current states, until one minute ahead of the current simulation time. Figure 6 illustrates the given explanation.

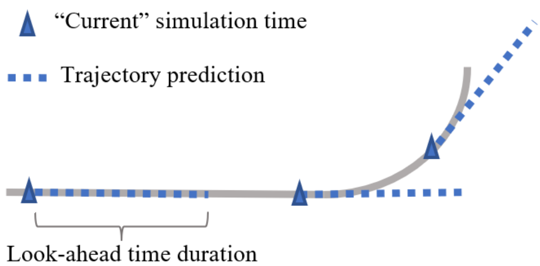


Fig. 6. Look-ahead time concept for state-based CD method exemplified

The first performance assessment approach for different look-ahead times will consider *true positives* (TP) and *false positives* (FP).

The TP happen when the predicted LoS is between the current simulation time and the look-ahead time, and when the actual LoS is between the current simulation time and the look-ahead time (plus a buffer of 10% the look-ahead time). The FP concern the detections that don't end up being real losses of separation (FP2) or that do but after the look-ahead time plus buffer extension (FP1). Figure 7 illustrates examples of both categories.

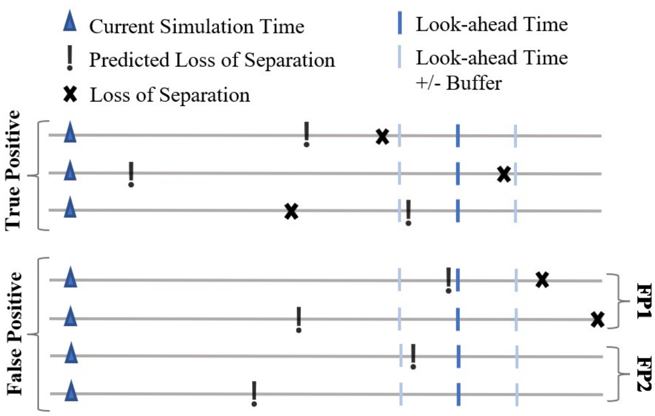


Fig. 7. First approach of look-ahead time

The first approach above illustrated brings one concern regarding the relative assessment accuracy since higher look-ahead time values will include the same conflicts as the lower look-ahead times. The distinction between these is not correctly evaluated in the first approach, named *flexible approach*. Thus, a second approach, named *strict approach* will be implemented in which *true positives*, *false positives* and *false negatives* (FN) are considered.

In this approach, TP exist when a predicted LoS is between the look-ahead time minus the buffer and the look-ahead time, while the actual LoS is between the buffer intervals around the look-ahead time. FP are considered when there is a predicted LoS but no actual LoS (FP2) or when there is an LoS but the prediction is not considered good enough (the predicted LoS is between the current simulation time and the inferior limit of the buffer, FP1). FN categorize the situations when the prediction of an LoS is within the look-ahead time minus buffer and look-ahead time, while the LoS happens before or after the buffer limits. Figure 8 illustrates the different parameters for the *strict approach*.

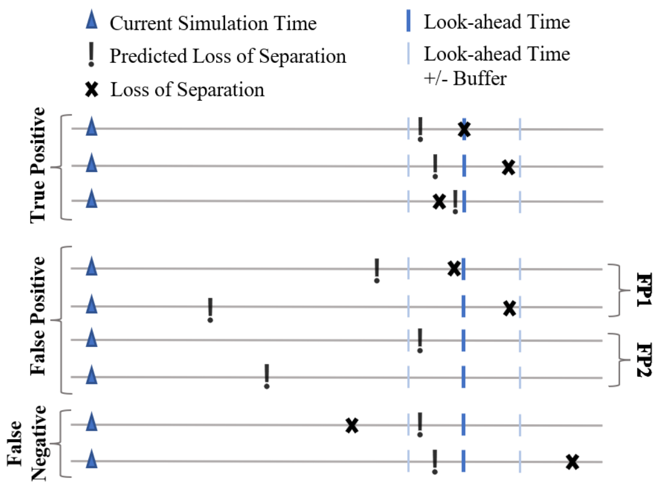


Fig. 8. Second approach of look-ahead time

The calculation of the results requires information from the data log output provided by BlueSky. The variables used from the data log are the *simulation time*, the *aircraft ID* of both aircraft involved in the conflict, and the *time to loss of separation*. Only the first data entry of a conflict is used and the same two aircraft can have a maximum of two distinct conflicts if these are spaced for more than 5 minutes, between the last alert of the first conflict and the first alert of the second one.

For each look-ahead time, two simulations are run, one with the wanted look-ahead time and one with 0s look-ahead time (reference value), stating an LoS detection and a conflict detection with a null look-ahead time are the same in the simulation conditions.

IV. Preliminary results & discussion

This section encompasses the results from the time shifting methods' assessment, in section IV-A, and the meteorological and traffic density bins, in section IV-B.

A. Time shifting Methods

Each time shifting technique has its characteristics and the results were obtained keeping these in mind. The tables below show the fitness results (F values), calculated as in the *genetic algorithm implementation*, for each one of the methods, followed by the comparison between them, using the best result for each technique.

The *random time adjustment* inherently has a random contribution, thus, five different runs were made to assess the coherence of the results. These are shown in Table III.

TABLE III
F VALUES FOR THE RANDOM TIME ADJUSTMENT METHOD

	1st run	2nd run	3rd run	4th run	5th run
F value	0.905	0.903	0.899	0.909	0.900

The *time compression* method does not have a random nature. Therefore, the results are the same for every run with the same parameters. In Table IV, the results for two t_c values are displayed.

TABLE IV
F VALUES FOR THE TIME COMPRESSION METHOD

	t_c = 0.8	t_c = 0.9
F value	0.893	0.888

The *genetic algorithm implementation* results were obtained to explore the influence of different parameters on the final result. The algorithm's stochastic characteristic is not explored in detail but the results are expected to be closer between them than in the *random time adjustment* due to the convergence characteristic of GAs. The results are below in Table V.

TABLE V
F VALUES FOR GENETIC ALGORITHM IMPLEMENTATION

Number of iterations	Crossover probability [%]	Mutation probability [%]	F value
20	75	3	0.9243
		8	0.9293
	90	3	0.9287
		8	0.9296
50	75	3	0.9402
		8	0.9338
	90	3	0.9455
		8	0.9367

Table VI shows the best simulations for each time shifting method and includes the analysis of the different parameters organized in distinct sub-tables, each presenting, independently, the absolute and percentage distribution of the total encounter number.

TABLE VI
TIME SHIFTED SCENARIOS WITH DIFFERENT TECHNIQUES FOR ACTUAL DATA

	Time Compression	**Random Time Adjustment**	**Genetic Algorithm**
Number of encounters	7063	5405	5526
Horizontal Distance			
0nm to 5nm	1068 15.12 %	934 17.28 %	1068 19.33 %
5nm to 10nm	1500 21.24 %	1095 20.26 %	1102 19.94 %
10nm to 15nm	1532 21.69 %	1146 21.20 %	1088 19.69 %
15nm to 20nm	1480 20.95 %	1123 20.78 %	1177 21.30 %
20nm to 25nm	1483 21.00%	1107 20.48 %	1091 19.74 %
Vertical Distance			
0ft to 500ft	3893 55.12 %	2965 54.86%	3101 56.12 %
500ft to 1000ft	342 4.84 %	299 5.53 %	317 5.74 %
1000ft to 2000ft	675 9.56 %	540 9.99 %	521 9.43 %
2000ft to 3000ft	638 9.03 %	472 8.73 %	497 8.99 %
3000ft to 4000ft	498 7.05 %	391 7.24 %	368 6.66 %
4000ft to 5000ft	1017 14.40 %	738 13.65 %	722 13.06 %
Relative Heading			
0º to 30º	2387 33.80 %	1850 34.23 %	1922 34.78 %
30º to 60º	1233 17.46 %	950 17.58 %	964 17.44 %
60º to 90º	1100 15.57 %	867 16.04 %	826 14.95 %
90º to 120º	615 8.71 %	477 8.82 %	550 9.95 %
120º to 150º	578 8.18 %	426 7.88 %	412 7.46 %
150º to 180º	1150 16.28 %	835 15.45%	852 15.42 %
Encounter Geometry			
Level-Level	1763 24.96 %	1331 24.63 %	1327 24.02 %
Level-Transitioning	1606 22.74 %	1245 23.03 %	1249 22.60 %
Transitioning - Transitioning	3694 52.30 %	2829 52.34 %	2950 53.38 %

The comparison between the three methods' performance in detail is crucial to understand if the F values are representative of the desired transformation goals.

The *genetic algorithm implementation* is the best technique to achieve a similar distribution to the filed data distribution, followed by the *random time adjustment* and the *time compression*, respectively. This evaluation comes from the F values analysis but also the distributions' comparison. The *time compression* has a *Number of encounters* clearly unlike all the others and the lowest percentage value for the 0-5NM parameter. This parameter along with the *Encounter Geometry* parameters were the critical parameters. However, the *Encounter Geometry* parameters did not reach the desired values for any of the techniques. Therefore, the analysis is based on F values and the 0-5NM *Horizontal Distance* bin.

However, the computational power required by the GA method is inefficient when several scenarios are considered. Therefore, to allow for some human bias correction in a feasible way, the *random time adjustment* was used for all the scenarios below.

B. Meteorological and Traffic Density Bins

The meteorological and traffic density bins were identified and a sample of those was selected for the further development of this research. Below, firstly, it is possible to acknowledge the bins' upper and lower bounds, as well as some clarity on the metrics used. Secondly, the bins' total size and the sample specifications are made available.

1) Bins' Upper and Lower bounds: For the traffic density bins, Table VII shows the upper and lower bounds for the three bins. It should be mentioned these are for a full day of data.

TABLE VII
TRAFFIC DENSITY BINS' UPPER AND LOWER BOUNDS

	Lower Bound [Nº of aircraft]	Upper Bound [Nº of aircraft]
Light	3277	6351
Medium	6352	7287
High	7288	8493

For the meteorological analysis, each F value per hour and lat-long grid cell is assessed. Table VIII shows the metrics to classify the F value in the L, M, and H categories. Even though Table VIII shows that the L category has a lower bound of $F = 4$, the F values have a broader range, reaching negative values. Every lat-long grid data has these lower values, regardless falling into the *Light*, *Medium*, or *High* bin. However, the number of these $F < 4$ values is unpredictable and can't be considered in the metrics defined in Table IX. For the *Light*, *Medium*, and *High* bins the metric is shown in Table IX, with $L_c \geq 0$, $M_c \geq 0$ and $H_c \geq 0$ being the highest hourly count (out of the six-hour simulation) of, respectively, L, M, and H. The balance from matching the data obtained and the data history resulted in Table VIII and Table IX.

TABLE VIII
AUXILIARY VARIABLES' BOUNDS

L	M	H
$4 < F \leq 12$	$12 < F \leq 20$	$F > 20$

The conditions in Table IX follow a *elif* condition (e.g. the *Medium* conditions are only evaluated if the data is not categorized as *Light* beforehand).

TABLE IX
METEOROLOGICAL BINS' UPPER AND LOWER BOUNDS

	Bounds
Light	$max(M_c) \leq 2 \wedge max(H_c) = 0 \wedge max(L_c) \leq 8$
Medium	$max(H_c) \leq 2 \vee (max(H_c) \leq 4 \quad \wedge max(M_c) > 0)$
High	$max(H_c) \geq 5$

2) *Bins' Size and Samples:* The meteorological bins match with the traffic density bins resulting in Table X. The fourth column specifies the samples picked from each bin, with the format "day/month/year" where AM stands for the 8 AM to 2 PM period, while the PM stands for the 2 PM to 8 PM period.

TABLE X
TOTAL BINS' SIZE AND SAMPLE PICK

Meteorological Bins	**Traffic Density Bins**	**Total Number of Elements**	**Random Sample of 4 Elements**
L	L	140	21/02/2017 - AM
			21/03/2017 - AM
			24/03/2018 - PM
			02/12/2018 - PM
	M	265	29/03/2017 - AM
			30/03/2017 - PM
			14/10/2017 - AM
			27/03/2019 - PM
	H	400	11/07/2018 - AM
			08/09/2018 - PM
			02/05/2019 - PM
			19/09/2019 - AM
M	L	252	04/01/2017 - PM
			11/01/2017 - AM
			04/11/2017 - AM
			02/11/2019 - AM
	M	391	08/02/2018 - AM
			31/10/2018 - AM
			19/12/2018 - AM
			22/12/2019 - AM
	H	524	24/08/2017 - AM
			01/10/2017 - AM
			07/09/2018 - PM
			21/07/2019 - AM
H	L	78	04/03/2017 - AM
			20/11/2018 - AM
			27/01/2019 - PM
			09/03/2019 - AM
	M	74	30/04/2018 - PM
			12/05/2018 - AM
			25/04/2019 - AM
			08/11/2019 - PM
	H	66	07/08/2018 - PM
			14/10/2018 - PM
			14/06/2019 - AM
			25/10/2019 - PM

V. RESULTS

The results showcase the two approaches mentioned before: the *flexible approach* in section V-A, followed by the *strict approach* in section V-B. The results display the trends for each look-ahead time, by calculating the average and standard deviation of each subset. The results for the thirty-six simulations enumerated in section IV-B are structured, in some cases, by category (traffic density or meteorological conditions).

The results make use of confusion matrix parameters, as mentioned before, *true positives* (TP), *false positives* (FP) and *false negatives* (FN).

A. Flexible Approach

The global performance of the thirty-six scenarios is pictured in Figure 9, where the performance parameters TP and FP (FP1, FP2) are displayed for each look-ahead time. The performance is in a ratio calculated by the number of conflicts for each parameter over the sum of $TP + FP$ since the focus is on the performance and not the total number of conflicts.

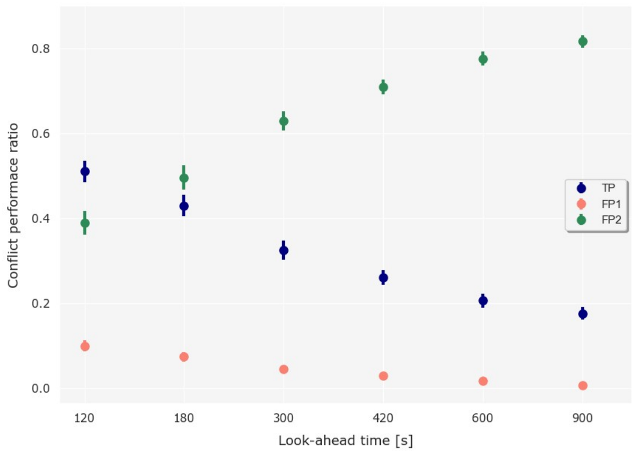


Fig. 9. Global performance (ratio) in *flexible approach*

The thirty-six scenarios were picked considering *Light*, *Medium* and *High* bins for the traffic density category and the meteorological conditions category. Figure 10 and Figure 11 illustrate this categorization, along with an additional distinction per flight phase, focusing on differentiating the climbing phase and descent phase from the cruise phase. The mentioned figures show the evolution of the conflict detection performance, for conflicts detected while in the cruise phase, for the different look-ahead times, and the different bins.

Figure 12 and Figure 13 use the same framework, however representing the general performance for all flight phases.

An additional result arrangement was obtained to validate the scenarios selection and analyze the influence of these variables. All the previous information is provided after a ratio calculation to facilitate visual comparison. Nevertheless, Figure 14 and Figure 15 present the results in absolute value to allow for validation of the scenario classification and additional conclusions.

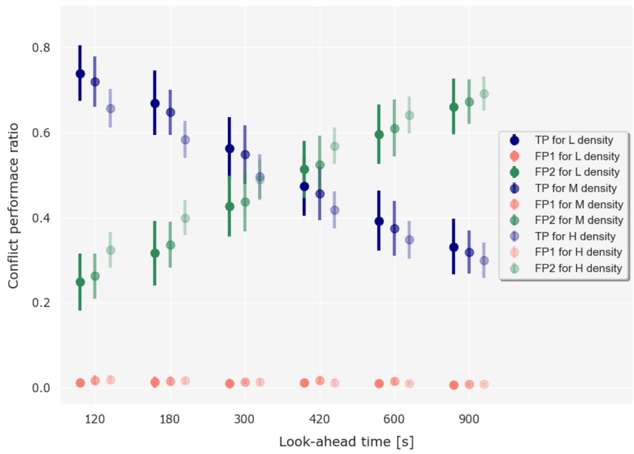


Fig. 10. Cruise phase performance (ratio) in *flexible approach* for traffic density bins L, M, and H

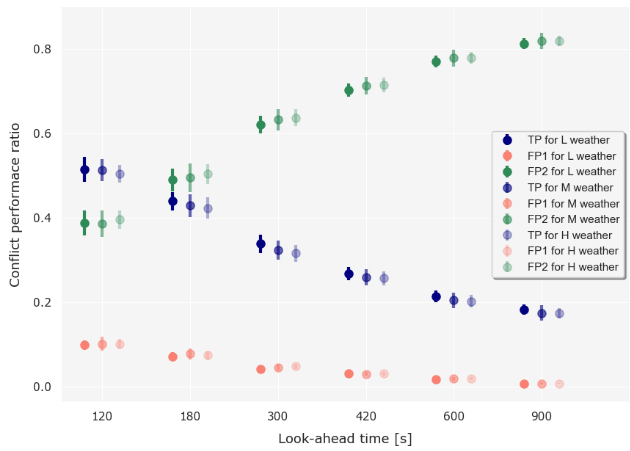


Fig. 13. Performance (ratio) in *flexible approach* for meteorological conditions bins L, M and H

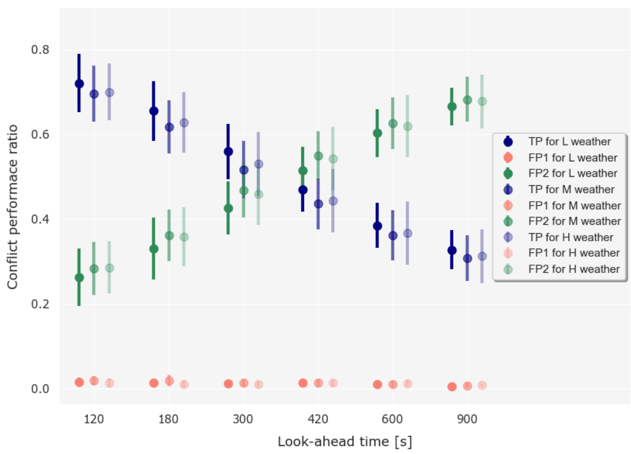


Fig. 11. Cruise phase performance (ratio) in *flexible approach* for meteorological conditions bins L, M, and H

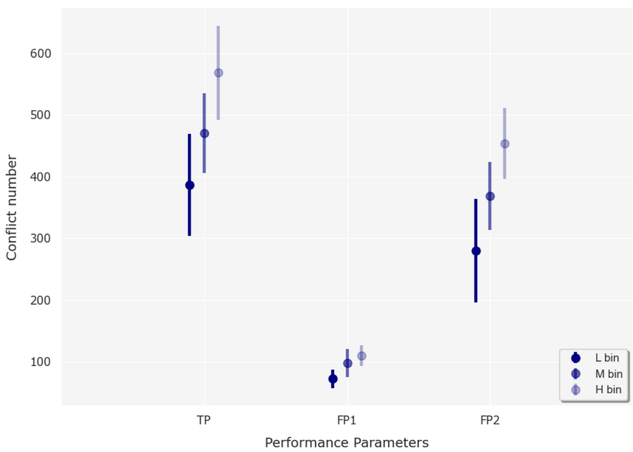


Fig. 14. Performance for 120s look-ahead time (absolute value) in *flexible approach* for traffic density bins L, M and H

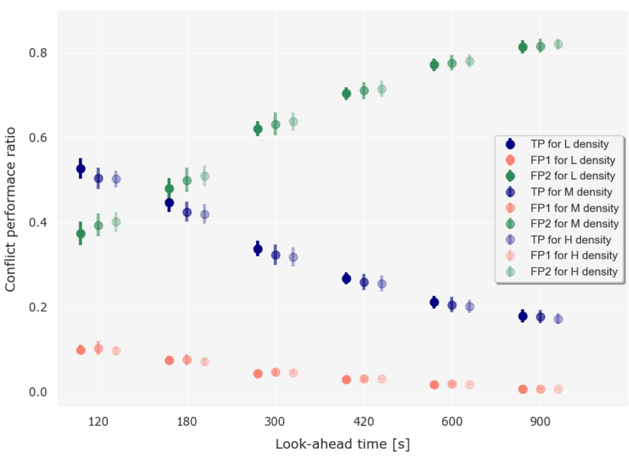


Fig. 12. Performance (ratio) in *flexible approach* for traffic density bins L, M and H

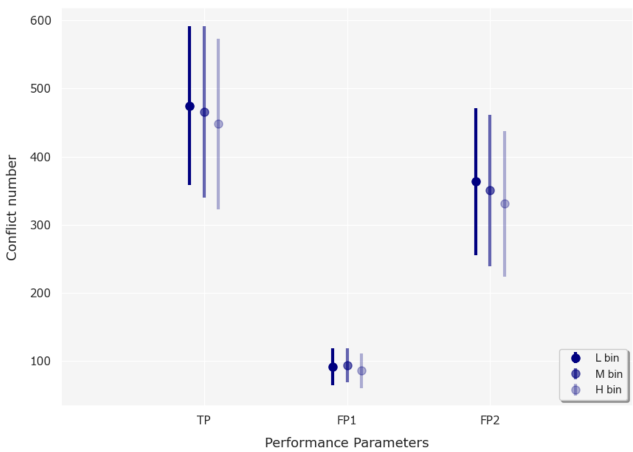


Fig. 15. Performance for 120s look-ahead time (absolute value) in *flexible approach* for meteorological conditions bins L, M and H

B. *Strict Approach*

The global performance for the second approach is illustrated in Figure 16. The performance for cruise phase conflicts is in Figure 17 and Figure 18 (with the category bins).

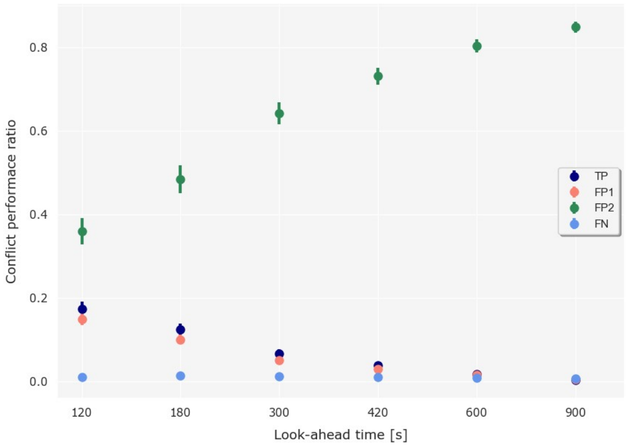


Fig. 16. Global performance (ratio) in *strict approach*

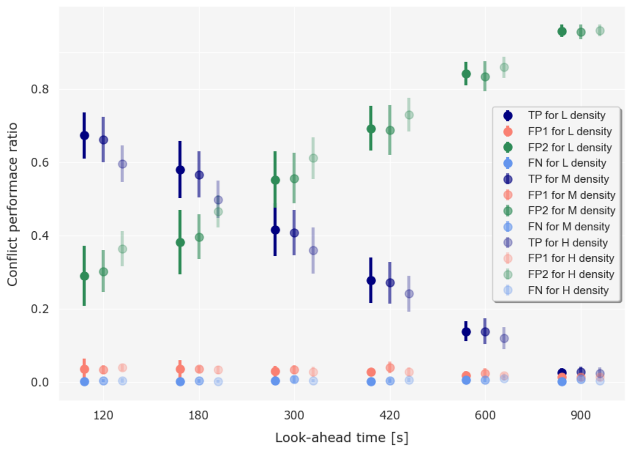


Fig. 17. Cruise phase performance (ratio) in *strict approach* for traffic density bins L, M, and H

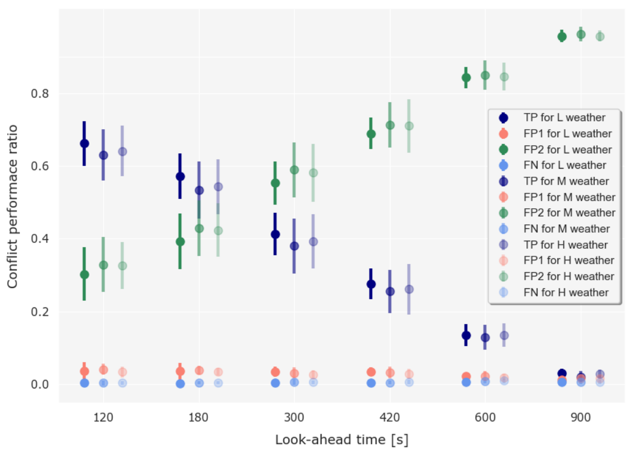


Fig. 18. Cruise phase performance (ratio) in *strict approach* for meteorological conditions bins L, M, and H

The results considering the category bins for the overall performance are in Figure 19 and Figure 20 for the traffic density and meteorological conditions bins, respectively.

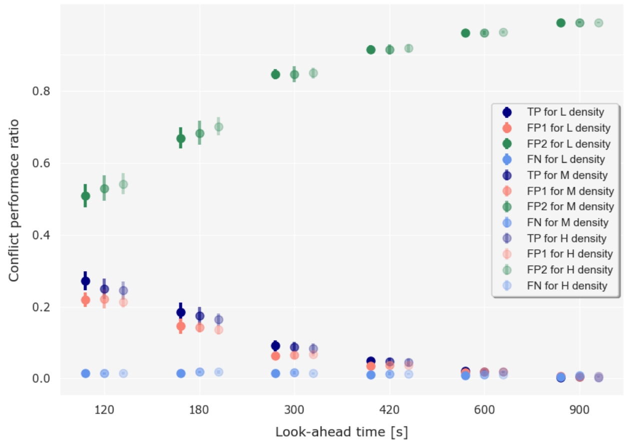


Fig. 19. Performance (ratio) in *strict approach* for traffic density bins L, M and H

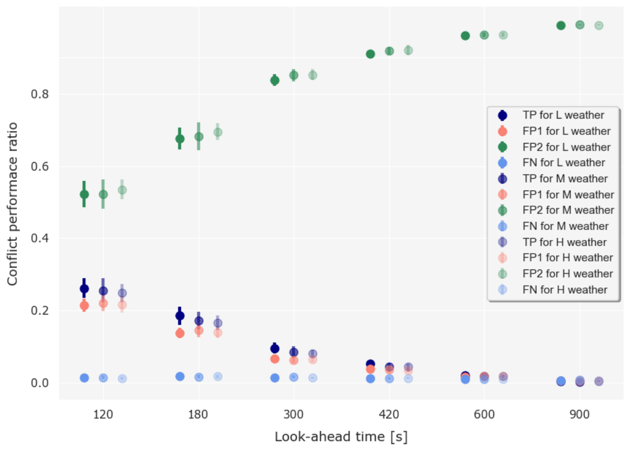


Fig. 20. Performance (ratio) in *strict approach* for meteorological conditions bins L, M and H

VI. Discussion

The results, in Figure 9, show the evolution for different look-ahead times, looking mostly at the average of the performance parameters. The standard deviation confirms the consistency of the conflict detection method's performance for the thirty-six scenarios since it can be considered low. The overall performance decreases with the increasing look-ahead time, something expected. However, the results show the ratio of *false positives* for three minutes (180s) is already higher than the *true positives* ratio. This outcome points to state-based conflict detection methods not having the characteristics required for conflict detection with a higher look-ahead time than 120s, while for cruise phase this performance stands for values below 300s. Nevertheless, it is worth mentioning the majority of FP are for LoS that did not happen, showing the state-based method predicts the real LoS quite accurately according to the metrics given (TP vs. FP1).

To explore the global performance in-depth the same metrics were applied, only to conflicts detected in cruise phase, resulting in Figure 10 and Figure 11. From these figures, it's possible to conclude conflicts in cruise phase contribute positively to the overall performance since the values of the ratios are higher for TP and lower for FP. The ratios for the cruise conflicts show the FP1 (the detected conflicts with a wrong time prediction) are considerably low values, also expected, considering the cruise phase is the steadiest of all flight phases. This explains, as well, the lower values for FP2 since there are fewer velocity and heading changes. The outcome of these results is this state-based conflict detection method is particularly effective for cruise phase, but less for climb and descent phases, especially for look-ahead times up to five minutes.

Figure 12 and Figure 13 show the same results, for all flight phases, also detailing the categories for traffic density and meteorological conditions, respectively. The results show *Light* bins are more favorable in these circumstances, but the difference between bins is not high (previously observed in the global performance). In addition, the impact different characteristics have on performance fades away with the increase in look-ahead time. Thus, the impact is mostly for the look-ahead times already considered valid for state-based methods. The cruise analysis is provided in the same format, highlighting similar trends to the global performance, mainly for the traffic density parameter.

The major difference between the categories' impact is meteorological conditions bins' performance values are more similar among them than when compared to the traffic density bins. The meteorological impact is not as noticeable as the traffic density's impact, potentially due to the bins' characteristics, since a high bin can have more severe weather conditions restricted to a specific area, allowing traffic to be normal in the surroundings. This is explored in Figure 14 and Figure 15 where the same results are shown but considering the absolute conflict number, for 120s look-ahead time, specifically since this is the look-ahead time marker with the highest contrast. Here, the difference is clearer since the traffic density conflict number increases visibly with the bin rating, as opposed to the meteorological conditions bins that have a conflict decrease but very tenuous. The escalating conflict number for the traffic density bins was expected, due to a rise in the aircraft number and air space complexity. The decline in the conflict number for the meteorological conditions was not anticipated.

The results for the *strict approach*, in Figure 16, show the same consistency of the conflict detection method's performance for the thirty-six scenarios. The overall performance decreases with the increasing look-ahead time, as before. Nevertheless, the results are quite alarming since the performance is very low even for the lowest look-ahead time value, 120s. From this, the conclusions are the accuracy of the predictions of real LoS is approximately 50% for each specific look-ahead time, according to the given metrics (TP and FP1 have approximately the same value).

Figure 17 and Figure 18 demonstrate the vast majority of the real LoS are correctly detected during cruise phase, making clear the contrast between the TP and FP1. In addition, as opposed to the situation in all flight phases, the ratio of TP is superior to FP, for small look-ahead times of 120s and 180s. The bin trends present in the results are similar to the previous approach, hence, no further remarks are made on this topic.

The results for *flexible approach* and *strict approach* were presented. There are some similarities such as the FP2 values are common for both approaches since the scenarios used were the same and the FP2 represent the conflicts that were not a real LoS. Thus, the FP2 in one approach are the same in the other approach.

The motive behind using two different approaches is evident in the data. The global performance results for the *flexible approach* show a considerable difference between the TP and FP1 ratios, suggesting from the real LoS there are at least 5x more TP than FP1 (e.g. for the highest look-ahead time, 900s). However, the TP values do not represent the accuracy at detecting conflicts 900s before the LoS but possibly for lower look-ahead times. The proof of this is the data for the *strict approach* where the TP and FP1 have both values close to zero. Thus, the performance of conflict detection methods as a function of look-ahead time depends highly on the metrics applied to assess it.

VII. Conclusion

The purpose of this paper focused on studying the performance of a state-based conflict detection method. The research to do this included recorded flight data processing (e.g. outlier elimination, filtering) and its manipulation to eliminate the inherent human bias. Time shifting was the technique implemented and from *time compression*, *random time adjustment*, and the *implementation of a genetic algorithm*, the last one proved to be the more effective. However, the random time adjustment technique was the chosen one, aiming for a more efficient process with the thirty-six scenarios.

The performance for different look-ahead times revealed itself relative after considering two distinct metrics to evaluate it. For more flexible metrics, the state-based conflict detection method is adequate for look-ahead times up to 120s, while for stricter metrics no considered look-ahead time value is fit enough for a good conflict detection performance. For both approaches, performance for cruise phase conflicts is higher than the overall performance, expected by the inherent characteristics of this phase with fewer heading and velocity changes.

For the secondary variables, traffic density's impact is consistent and shows that an increase in traffic density translates into a decrease in the state-based conflict detection method's performance. The meteorological conditions' influence is not as notorious, thus no conclusions can be withdrawn, even though a similar trend is observed.

This research focused on contributing to the Air Traffic Management community by working with recorded air traffic data and realistic simulation settings. To complement this research, an intent-based conflict detection method should be used, aiming to get closer to the used conflict detection method by air traffic controllers, and metrics fitter to the impact conflict detection tools have on the ones who use them should be considered.

References

[1] Sameer Alam et al. "Evolving air traffic scenarios for the evaluation of conflict detection models". In: *6th EUROCONTROL Innovative Research Workshop and Exhibition: Disseminating ATM Innovative Research* (2007), pp. 237–245.

[2] *Archive data for R&D - Eurocontrol*. https://ext.eurocontrol.int/prisme_data_provision_hmi/. Accessed: 2021-03.

[3] J. E. Beasley, H. Howells, and J. Sonander. "Improving short-term conflict alert via tabu search". In: *Journal of the Operational Research Society* 53.6 (2002), pp. 593–602.

[4] William Chan, Ralph Bach, and Joseph Walton. "Improving and validating CTAS performance models". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit* August (2000).

[5] Joost Ellerbroek and Jacco M Hoekstra. "BlueSky ATC Simulator Project: an Open Data and Open Source Approach Three-Dimensional Airborne Separation Assistance Displays View project BlueSky-Open source ATM simulator View project BlueSky ATC Simulator Project: an Open Data and Open Source Approach". In: *seventh International Conference for Research on Air Transport (ICRAT)* (2016).

[6] *ERA5 hourly data on pressure levels from 1979 to present [online]*. https://cds.climate.copernicus.eu/cdsapp#!/dataset/10.24381/cds.bd0915c6?tab=overview. Accessed: 2021-12.

[7] *Eurocontrol Aviation Intelligence Portal - Data Download (Info) [online]*. https://ansperformance.eu/reference/dataset/en-route-atfm-delay-aua/. Accessed: 2021-11.

[8] *Eurocontrol Aviation Intelligence Portal [online]*. https://ansperformance.eu/data/. Accessed: 2021-11.

[9] Andrew J Fabian et al. "Design and Performance of an Improved Genetic Algorithm Implementation for Time-Shifted Air Traffic Scenario Generation". In: (2016).

[10] *Flightradar24 [online]*. https://www.flightradar24.com/data. Accessed: 2021-11.

[11] *Github Repository [online]*. https://github.com/lunajuliao/CDperformance. Accessed: 2022-05.

[12] Sehchang Hah, Ben Willems, and Randy Phillips. "The effect of air traffic increase on controller workload". In: *Proceedings of the Human Factors and Ergonomics Society [e-journal]* (October, 2006), pp. 50–54.

[13] H. Hersbach et al. "ERA5 hourly data on pressure levels from 1979 to present". In: *Copernicus Climate Change Service (C3S) Climate Data Store (CDS)* (2017).

[14] J M Hoekstra. "Free Flight in a Crowded Airspace?" In: *Air Transportation Systems Engineering* June (2001), pp. 533–545.

[15] ICAO. *Doc 4444 - Air Traffic Management - Procedures for Air Navigation Services*. 2016, pp. 5.1–5.81.

[16] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, 1992.

[17] Todd A. Lauderdale, Andrew C. Cone, and Aisha R. Bowe. "Relative significance of trajectory prediction errors on an automated separation assurance algorithm". In: *Proceedings of the 9th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2011* (2011), pp. 381–390.

[18] William P. Niedringhaus. "Solution Complexity Metrics". In: *1998 Guidance, Navigation, and Control Conference and Exhibit* (1998), pp. 30–51.

[19] *OpenSky Network [online]*. https://opensky-network.org/datasets/metadata/. Accessed: 2021-05.

[20] Mike M. Paglione, Robert D. Oaks, and J. Scott Summerill. "Time shifting air traffic data for quantitative evaluation of a conflict probe". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit* (2003).

[21] Russell A Paielli and Heinz Erzberger. "Tactical Conflict Detection Methods for Reducing Operational Errors". In: *Air Traffic Control Quarterly* 13.1 (2005), pp. 83–106.

[22] Russell A. Paielli. "Empirical Test of Conflict Probability Estimation". In: *2nd USA/Europe Air Traffic Management R&D Seminar (ATM-98), Orlando, Florida* (1998).

[23] Rhys Parfitt, Arnaud Czaja, and Hyodae Seo. "A simple diagnostic for the detection of atmospheric fronts". In: *Geophysical Research Letters* 44.9 (2017), pp. 4351–4358.

[24] Matthias Schäfer et al. "Bringing up OpenSky: A large-scale ADS-B sensor network for research". In: *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. 2014, pp. 83–94.

[25] Emmanuel Sunil et al. "Three-dimensional conflict count models for unstructured and layered airspace designs". In: *Transportation Research Part C: Emerging Technologies* 95.April (2018), pp. 295–319.

[26] David Thipphavong. "Analysis of a multi-trajectory conflict detection algorithm for climbing flights". In: *9th AIAA Aviation Technology, Integration and Operations (ATIO) Conference, Aircraft Noise and Emissions Reduction Symposium (ANERS)* (2009), pp. 1–13.

[27] *Time and Date AS [online]*. https://www.timeanddate.com/weather/netherlands/amsterdam/historic. Accessed: 2021-10.

[28] *UK Civil Aviation Authority [online]*. https://www.caa.co.uk/Commercial-industry/Airspace/Air-traffic-control/Air-navigation-services/UK-Wake-Turbulence-Categories/. Accessed: 2021-10.

[29] T. Zeh et al. "Prediction of the propagation of trajectory uncertainty for climbing aircraft". In: *AIAA/IEEE Digital Avionics Systems Conference - Proceedings* (2020).

III

Closure

4
Conclusion

The ATM community aims to improve current conflict detection technologies to maximize the airspace capacity. This is achieved by studying new technologies, evaluating the current ones, and discovering flaws, among other processes. The research work reached for studying the performance of a state-based conflict detection method, with the more realistic simulation characteristics. The research was guided to answer the three *research questions* mentioned before and finally achieve the *research objective*.

Firstly, the focus is on how to achieve the real-life conditions for the scenarios, thus, answering the question *"What are the characteristics of the most realistic (assumptions free) air traffic scenarios?"*. From the research done, to increment the veracity of the data and results, recorded flight data should be used with a short time step in order to allow the replication of the recorded trajectories accurately. OpenSky data was used to achieve this goal. In addition, one of the requisites to work with recorded air flight data and conflict detection assessments is eliminating the human bias imposed by solving and avoiding conflict actions. Time shifting is the concept applied to counteract this bias and three different approaches were studied: time random adjustment, time compression, and implementation of a genetic algorithm. The implementation of a genetic algorithm is a results-oriented technique and it achieves the best result out of the three strategies. Nevertheless, the second-best technique, random time adjustment, was the one selected due to its computational efficiency.

Secondly, the metrics considered to evaluate the performance of a conflict detection method have to be considered thoughtfully. To answer the question *"Which metrics and parameters to use when calculating the performance of conflict detection methods?"*, two different approaches were considered. The metrics to classify a conflict detection method should be adjusted to the users' (air traffic controllers) feedback on effectiveness and workload. Hence, two approaches were defined, *flexible approach* and *strict approach*, aiming to characterize two evaluations almost opposed to each other that can cover all the scenarios in between. The *flexible approach* considers a *true positive*, thus a correct detection, for any conflict detected with the LoS happening between the current simulation time and the look-ahead time plus a buffer, while the *strict approach* only contemplates a *true positive* when the conflict detection and the LoS happen both in an interval around the set look-ahead time. This distinction is highly important for higher look-ahead times (e.g. 600s) since, for the *flexible approach*, conflicts that happen at the 120s contribute to the good performance of the method, while this attribution does not mean the method can detect conflicts correctly 600s ahead. The *strict approach* offers the opposite perspective, only considering conflicts correctly detected around 600s.

Finally, to answer the main research question *"What is the global performance of BlueSky conflict detection methods considering different look- ahead times?"*, a more complex analysis is done with different steps, involving different topics. While assessing the performance of the state-based conflict detection method, traffic density, meteorological conditions, and flights phases were considered. The impact of traffic density on conflict detection performance is expected, with a higher traffic density translating to lower overall performance. The meteorological conditions' impact appears to follow a similar trend but no conclusions are withdrawn from the results, due to the lack of distinction between the results for *Light*, *Medium*, and *High* bins. The global performance for the *flexible approach* revealed state-based conflict detection methods are adequate for look-ahead times up to 120s while the *strict approach* did not find any suitable look-ahead times for the use of state-based conflict detection methods. These considerations are not representative of each one of the three flight phases (climb phase, cruise phase, and descent phase), since the cruise phase performance is considerably higher than the other two phases.

The research objective below, also stated at the beginning of this research, was achieved by obtaining very different types of data, applying different processes and techniques, and always building this project to contribute to the ATM community.

> “To contribute to the comprehension of Air Traffic Management tools’ accuracy by conducting an analysis on different scenarios where conflict detection methods performance is assessed and correlated with the set look-ahead time.”

5

Recommendations for future work

This project leaves an open path to continue the research done, aiming to contribute more to the Air Traffic Management community as many others do. In this section, some enhancement remarks will be laid out as well as additional features that can be implemented on top of the work done or could be the next step in new independent research.

Initially, to refine the results by eliminating any possible bias and to enhance the process efficiency, the remarks below should be considered:

- Obtain data for a wider geographical area: increases the flights with all flight phases, namely increases the percentage of flights going through cruise.
- Obtain and match ADS-B data from different data sources - The ADS-B raw data from OpenSky can not be used directly in air traffic simulations due to noise effect and potentially resolution errors (the resolution is one second). Thus, matching different data sources could boost the accuracy of the trajectories.
- Create data documents with a higher Δt - Obtaining data with increased resolution allows for more accurate data. Therefore, the necessity for input scenario indications with a $\Delta t = 1$s is not as necessary and if this is the case, the efficiency of the processing increases. The conflict data log could also be considered with a time step of 2s, looking for efficiency improvements.

After taking into consideration the potential improvements in the completed research, other projects could ameliorate and complement the current conclusions.

- Using an intent-based conflict detection method - the state-based method is not the method used by air traffic controllers nowadays. The intent-based conflict detection method resembles more closely a real conflict detection simulation. Thus, its results would be more accurate and useful for the community
- Accommodate performance parameters with real ATC controllers considerations - correlate the metrics with the workload of the air traffic controllers.

IV

Appendices

A

OpenSky Data Adjustments and Assumptions

This chapter concerns the necessary modifications to the flight data, already mentioned in the second scientific paper. Each correction or assumption will be further explained and exemplified.

The data incoherence is considered to be a consequence of the nature of the data since OpenSky makes available ADS-B data and the coordination of all the receptors together with any possible noise can generate these bugs. In addition, this ADS-B format provides data with time resolution to the second, while air traffic data can change significantly in a tenth of a second.

A.1. Callsign bug

OpenSky uses the callsign as a flight ID and it's the only flight identifier provided since the icao24 code can be the same for the same aircraft and different flights. Thus, the callsign is the parameter used to make the distinction between flights. A problem arises when some data is incoherent as it is possible to see in Figure A.1, where the correct callsign "RJA262" appears as "BY G F" for two consecutive data entries. This information can not be matched to the correspondent flight, so it's deleted. Any identified flight with a data length inferior to 30 seconds is not considered, eliminating these situations (usually up to 5-6 seconds).

Time	Icao24	Latitude [°]	Longitude [°]	Velocity [$\frac{m}{s}$]	Vertical Rate [$\frac{m}{s}$]	Baroaltitude [m]	Geoaltitude [m]	Callsign
1535876002	740827	52.13543	3.90305	117.89085	-0.32512	12489.18	13014.96	RJA262
1535876003	740827	52.12822	3.92518	117.89085	-0.32512	12496.80	13022.58	RJA262
1535876004	740827	52.12822	3.92518	117.89085	-0.32512	12496.80	13022.58	RJA262
1535876005	740827	52.12822	3.92518	117.89085	-0.32512	12489.18	13022.58	BY G F
1535876006	740827	52.12509	3.93471	117.89085	0.00000	12489.18	13014.96	BY G F
1535876007	740827	52.12417	3.93750	117.89085	-0.32512	12496.80	13014.96	RJA262
1535876008	740827	52.12319	3.94056	117.89085	-0.32512	12489.18	13014.96	RJA262
1535876009	740827	52.12217	3.94362	117.89085	-0.32512	12489.18	13014.96	RJA262

Figure A.1: Segment of flight *RJA262* - raw data columns from OpenSky - callsign highlight

The consequent database after this action is showcased in Figure A.2, where it's possible to see the time gap caused in the highlighted fragment.

Time	Icao24	Latitude [°]	Longitude [°]	Velocity [$\frac{m}{s}$]	Vertical Rate [$\frac{m}{s}$]	Baroaltitude [m]	Geoaltitude [m]	Callsign
1535876002	740827	52.13543	3.90305	117.89085	-0.32512	12489.18	13014.96	RJA262
1535876003	740827	52.12822	3.92518	117.89085	-0.32512	12496.80	13022.58	RJA262
1535876004	740827	52.12822	3.92518	117.89085	-0.32512	12496.80	13022.58	RJA262
1535876007	740827	52.12417	3.93750	117.89085	-0.32512	12496.80	13014.96	RJA262
1535876008	740827	52.12319	3.94056	117.89085	-0.32512	12489.18	13014.96	RJA262
1535876009	740827	52.12217	3.94362	117.89085	-0.32512	12489.18	13014.96	RJA262

Figure A.2: Segment of flight *RJA262* - data after callsign correction - time gap highlight

A.2. Duplicate data and interpolation

In the data obtained, consecutive data entries with different timestamps appear with the same latitude and longitude coordinates for a flying aircraft, which can not happen in real life. It happens and it's admissible if the aircraft in question is on the ground but this scenario does not interest this research so all situations go through the same process. Thus, the first duplicate data entry is kept but all other duplicates are eliminated, resulting in timestamp jumps of usually two to three seconds, but in some cases going up to twenty seconds. This situation is clear for the 2_{nd}, 3_{rd} and 4_{th} data entries in Figure A.3.

Time	Icao24	Latitude [°]	Longitude [°]	Velocity [$\frac{m}{s}$]	Vertical Rate [$\frac{m}{s}$]	Geoaltitude [m]	Callsign
1535876396	4caa5a	51.42472	5.34599	74.34561	-4.22656	160.020	RYR41UV
1535876397	4caa5a	51.42584	5.34721	74.34561	-4.22656	152.400	RYR41UV
1535876398	4caa5a	51.42584	5.34721	74.63872	-3.57632	137.160	RYR41UV
1535876399	4caa5a	51.42584	5.34721	74.63872	-3.57632	137.160	RYR41UV
1535876400	4caa5a	51.42761	5.34920	74.63872	-4.22656	129.540	RYR41UV
1535876401	4caa5a	51.42784	5.34943	74.63872	-4.22656	129.540	RYR41UV
1535876403	4caa5a	51.42891	5.35065	74.63872	-3.25120	121.920	RYR41UV
1535876404	4caa5a	51.42947	5.35131	74.63872	-3.25120	121.920	RYR41UV

Figure A.3: Segment of flight *RYR41UV* - duplicate data and missing information highlight

An initial threshold for this jump was set for 8 seconds but it resulted in the exclusion of considerable significant data. Therefore, the threshold was raised to 30 seconds. The flight used as an example previously (*RJA262*) had, for example, a twenty consecutive data points gap.

Nevertheless, looking at the example these data manipulations would result in a time gap of two seconds. In addition, the data obtained has breaches of information in time, incremented by the manipulations (e.g. the callsign case). Hence, data interpolation helps to bridge this situation. The interpolation was implemented to obtain the flight information per second, for every flight. Figure A.4 shows the same time frame but with the changes implemented.

Time	Icao24	Latitude [°]	Longitude [°]	Velocity [$\frac{m}{s}$]	Vertical Rate [$\frac{m}{s}$]	Geoaltitude [m]	Callsign
1535876396	4caa5a	51.42472	5.34599	74.34561	-4.22656	160.020	RYR41UV
1535876397	4caa5a	51.42584	5.34721	74.34561	-4.22656	152.400	RYR41UV
1535876398	4caa5a	51.42643	5.34787	74.44331	-4.22656	144.780	RYR41UV
1535876399	4caa5a	51.42702	5.34854	74.54101	-4.22656	137.160	RYR41UV
1535876400	4caa5a	51.42761	5.34920	74.63872	-4.22656	129.540	RYR41UV
1535876401	4caa5a	51.42784	5.34943	74.63872	-4.22656	129.540	RYR41UV
1535876402	4caa5a	51.42838	5.35004	74.63872	-3.73888	125.730	RYR41UV
1535876403	4caa5a	51.42891	5.35065	74.63872	-3.25120	121.920	RYR41UV
1535876404	4caa5a	51.42947	5.35131	74.63872	-3.25120	121.920	RYR41UV

Figure A.4: Segment of flight *RYR41UV* - data after duplicate data correction and interpolation - interpolation highlight

A.3. Altitude and vertical rate coherence

OpenSky provides altitude information in two different formats: *barolatitude* and *geoaltitude*. Both formats present discontinuities and don't resemble a continuous real trajectory, which is a data resolution and noise issue. In addition, in some cases, the vertical rate is not consistent with the vertical position progression (e.g. for constant altitude values there are non-zero vertical rate values, as shown in Figure A.5 for flight *RYR41UV*).

The coherence between velocity and coordinates is crucial to traffic simulations, especially considering the goal of this research: study the performance of a *state-based* conflict detection software.

Time	Icao24	Latitude [°]	Longitude [°]	Velocity [$\frac{m}{s}$]	Vertical Rate [$\frac{m}{s}$]	Baroaltitude [m]	Geoaltitude [m]	Callsign
1535876396	4caa5a	51.42472	5.34599	74.34561	-4.22656	38.100	160.020	RYR41UV
1535876397	4caa5a	51.42584	5.34721	74.34561	-4.22656	30.480	152.400	RYR41UV
1535876398	4caa5a	51.42584	5.34721	74.63872	-3.57632	30.480	137.160	RYR41UV
1535876399	4caa5a	51.42584	5.34721	74.63872	-3.57632	30.480	137.160	RYR41UV
1535876400	4caa5a	51.42761	5.34920	74.63872	-4.22656	22.860	129.540	RYR41UV
1535876401	4caa5a	51.42784	5.34943	74.63872	-4.22656	11582.400	129.540	RYR41UV
1535876403	4caa5a	51.42891	5.35065	74.63872	-3.25120	15.240	121.920	RYR41UV
1535876404	4caa5a	51.42947	5.35131	74.63872	-3.25120	7.620	121.920	RYR41UV

Figure A.5: Segment of flight *RYR41UV* - raw data columns from OpenSky - altitude and vertical rate coherence highlight, outlier highlight

A solution was implemented, starting by filtering the altitude data and obtaining then the vertical rate, assuming position data is more accurate. However, the resolution and noise issue was affecting the data to the point a filter couldn't be tuned for cruise and descending trajectories simultaneously (e.g. bridging the resolution and noise issue for cruise phase, led to descent trajectories with distinct slopes from the originals). Hence, a different solution to this problem arose since it is possible to do the reverse process and calculate the altitude making use of the vertical rate, obtaining this way coherent results for these parameters.

Thus, the vertical rate variable was evaluated but looking at Figure A.6 it is comprehensible that excluding the outliers was necessary. This exclusion was done before the interpolation since the interpolation could, in some cases, transform an outlier into a valid data point. It is worth mentioning a moving average (20 data entries) and a low-pass filter were applied to the vertical rate. This rectification was introduced due to the lack of realism of some trajectories, to smooth the recurrent abrupt changes in the vertical trajectories (accounting there are abrupt changes in ascending and descending geometries). The low-pass filter was built based on the *scipy.signal* library, in Python, with a 0.1Hz cut-off frequency, 1Hz sampling frequency and order 2 ButterWorth filter.

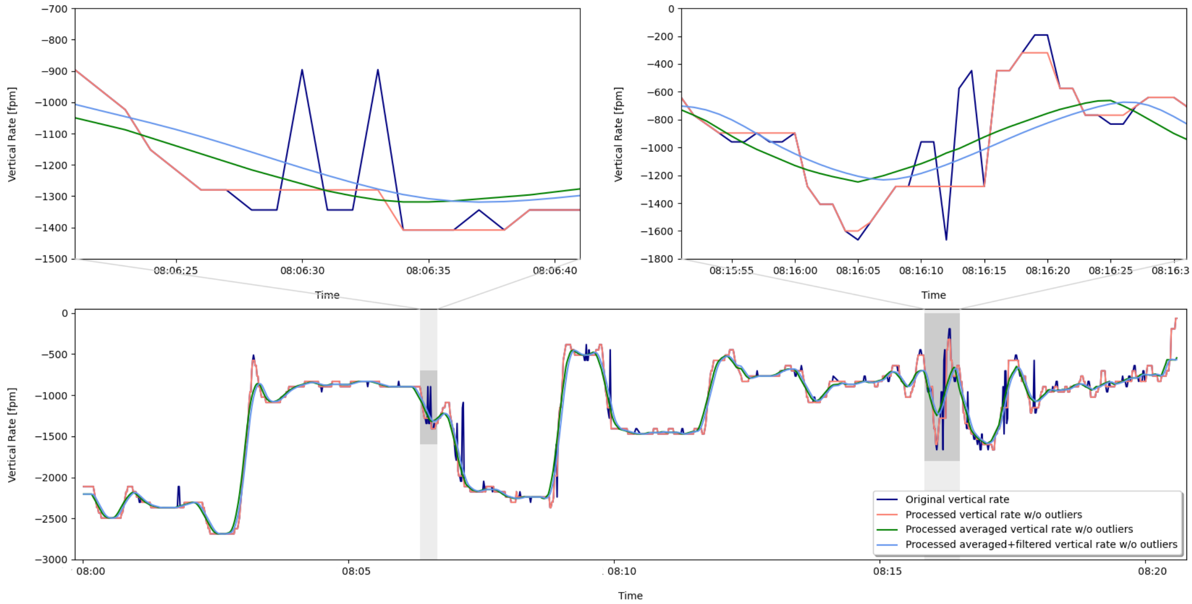


Figure A.6: Vertical rate before and after filter for *RYR41UV* flight

A new altitude was computed using the first *geoaltitude* data entry, for each flight, the obtained vertical rate for each time step, and the heading data (after processing). The heading data processing was not a critical step but it was done to improve the quality of the altitude data. The processing relied on low-pass filtering (first-order filter, with a 0.04Hz cut-off frequency and 1Hz sampling frequency) the *sin* and *cos* functions of the heading values since these are bound between $[0, 2\pi]$.

Figure A.7 shows the overall new vertical trajectory is coherent with the data made available and makes clear the difference to the original one.

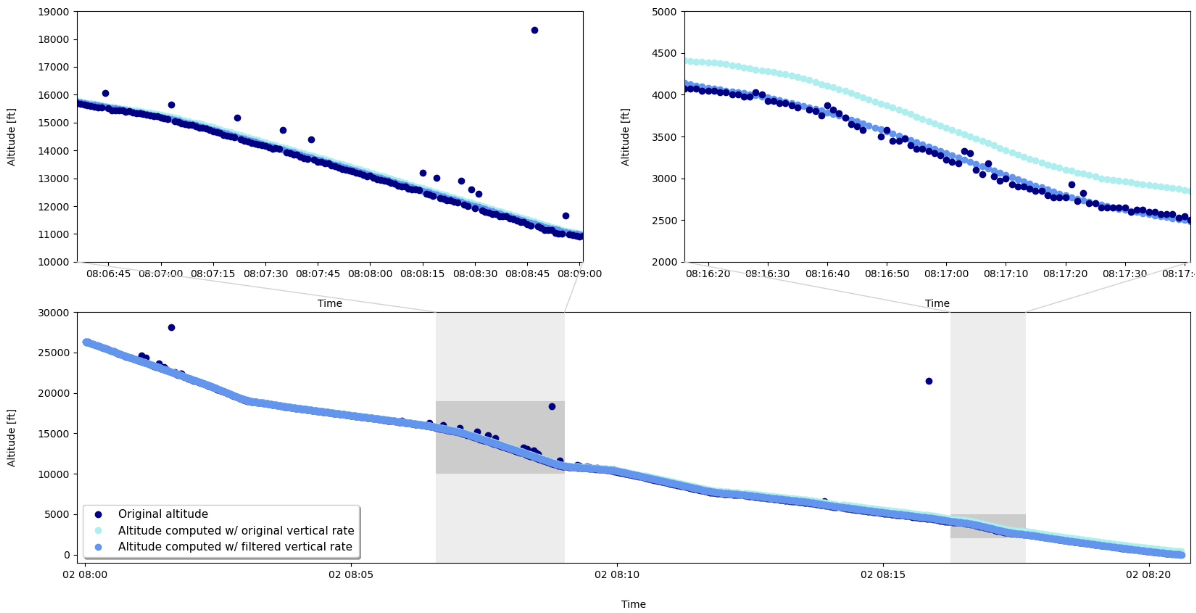


Figure A.7: Altitude for *RYR41UV* flight

.

A.4. Horizontal velocity and coordinates

The data for the horizontal axis, latitude and longitude coordinates, did not match the velocity given, made clear by the simulation in BlueSky where the phenomena in Figure A.8 would usually happen. The first scenario illustrates the ideal situation but the real scenario was composed of the second and third scenarios. These exemplify situations when the velocity given at each time step is not enough to reach the following position and then there is a forward jump (scenario 2) or when the velocity is too high for that time step (scenario 3) and the aircraft jumps backward to adjust to the position input.

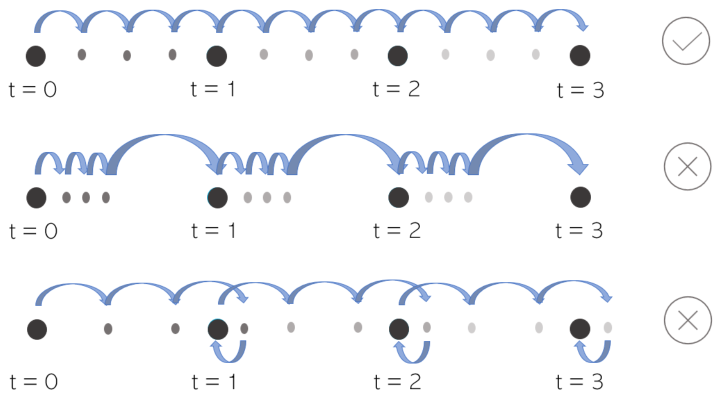


Figure A.8: Velocity and position mismatch

To rectify this, since the resolution and noise problem was not as alarming, a new velocity for each time step was calculated from the position data (the instantaneous velocity). Then, a moving average was computed from that velocity (distance between position data points). These steps made clear the disparity between the given and the calculated values. The last step consisted on smoothing, the same way as in the vertical velocity case, with a low-pass filter. In addition, to have more continuous heading transitions, this parameter went through the same low-pass filter smoothing (sine and cosine functions were used). The results are in Figure A.9.

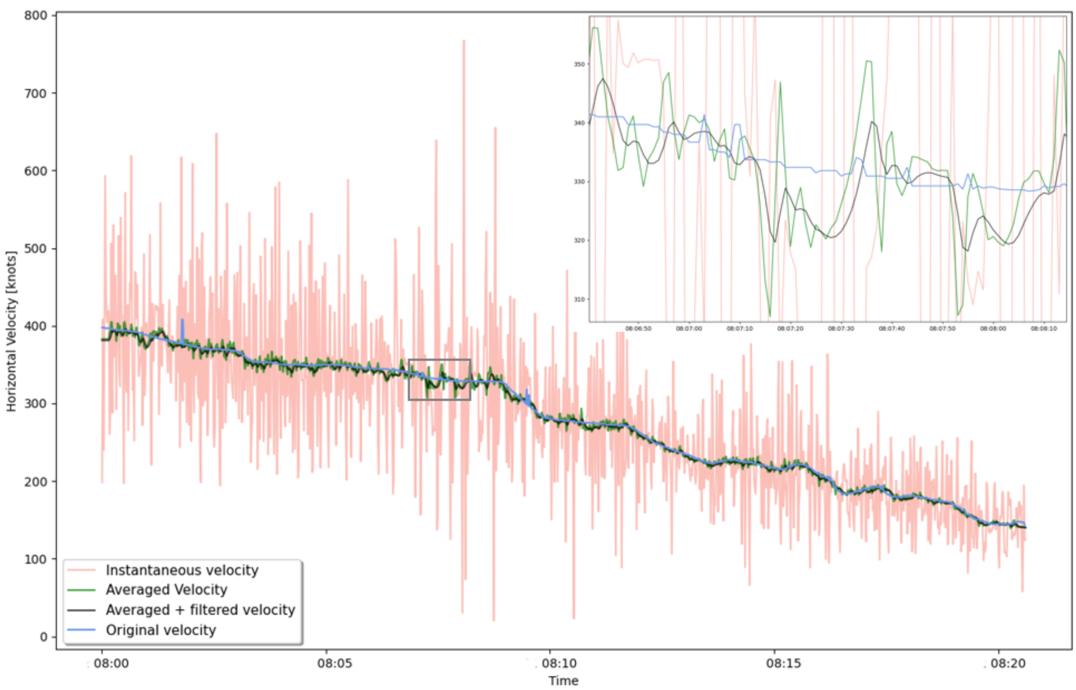


Figure A.9: Horizontal velocity for *RYR41UV* - for the different calculation steps

By looking at the mentioned figure, it stands out the result is very similar to the original given velocity, even more, that the given velocity's transitions are smoother than the calculated ones. Thus, understanding the difference between the obtained trajectories from the calculated velocity and the original velocity was an important step. The difference between the two was negligible, thus, the original velocity was the chosen one.

B

Genetic Algorithm

The genetic theory applied to computational algorithms was first introduced by Holland (1992, [4]). This theory that evolved into different genetic algorithms (GA) gets its fundamental principles from biologic phenomena and Charles Darwin's concept of natural selection. The concept behind the algorithm and its implementation are elaborated in chapter II, in both elements, but a detailed exposition of the implementation will is done in section B.1, followed by the validation used initially when the algorithm was built, in section B.2.

B.1. Implementation

The genetic algorithm was implemented in Python but did not use any libraries made available such as *GeneAI* or *gaframework* [6]. The algorithm was built from scratch, based on the code found in *Machine Learning Mastery* [5], since some particularities seemed incompatible with the libraries' features. The major issues were several long steps in the fitness function. In some situations, a time gap was required and, in order to make the process more efficient, multi-threading and multi-processing were integrated into the *blue_run()* function. This function is in the algorithm below and includes the time shift process, the BlueSky simulation, and the fitness value attribution.

Algorithm 1 Genetic algorithm main

Require: $N_{generation}$, $N_{individuals}$, N_{genes}, $P_{crossover}$,$P_{mutation}$
Ensure: $Best$

Initialize $pop \leftarrow$ with random normally distributed values
Initialize $Best \leftarrow$ with value 0
for var_{gen} **in** range($N_{generation}$) **do**
    $objective \leftarrow$ *blue_run(pop)* function outputs scores from the current generation
    $Best \leftarrow$ element with highest score
    $selected \leftarrow$ the three elements with the highest score (*selection()* function)
    $children \leftarrow$ two copies from the two best individuals
    $parents \leftarrow$ pairs of all combinations between the selected individuals
    for *c* **in** *crossover(parents)* **do**
        $children \leftarrow$ *mutation(c)* function output - join new individual
    end for
    $pop \leftarrow children$
end for
    return Best

The algorithm above is similar to most genetic algorithms and its name was attributed due to the inspiration on the biological evolution process. The *selection()*, *crossover()* and *mutation()* functions come directly from some scientific concepts.

The $blue\_run$() function's algorithm is revealed below with the the multi-threading implementation, followed by the *Scenario* thread implementation, where the multi-processing is evident by the use of *Popen()*.

Algorithm 2 $blue\_run()$ function

Require: pop
Ensure: $Results$

Initialize $scenarios \leftarrow$ empty list
for i **in** range(length(pop)) **do**
    $scenario \leftarrow$ *Scenario()* thread for i^{th} value
    scenario.start
    $scenarios \leftarrow$ join scenario in every iteration
end for
while None **in** *scenarios.result* **do**
    pass
end while
for i **in** range(length(pop)) **do**
    $Results \leftarrow$ join scenario in every iteration
end for
    return Results

Algorithm 3 $Scenario$ class - run segment

Require: *self*
Ensure: *self*

$SCN \leftarrow$ the time shifted scenario name ($time\_shift()$ output)
$self.p \leftarrow$ *Popen()* the BlueSky script with *SCN*
while None **in** *p flag* **do**
    pass
end while
$self.r \leftarrow$ *Popen()* the Evaluation script
while None **in** *r flag* **do**
    pass
end while
$self.result \leftarrow$ results from the evaluation script
    return *self*

B.2. Validation

A simple assessment was done to understand the behavior of the genetic algorithm concerning its structure. Thus, instead of having a time shift function, the BlueSky software, and a data log evaluation function, this algorithm aims to maximize (maximum 1) the average of an array with 100 random values $\in]0,1[$. At the same time, due to the similarity in the result type, some tests were run to compare the performance for different parameters such as crossover and mutation probabilities and number of iterations. Table B.1 shows the results for the different combinations of crossover and mutation probabilities for 50 iterations. Figures B.1, B.2, B.3 and B.4 show the evolution of each one of the simulations.

Table B.1: *F* values for genetic algorithm implementation

Number of iterations	Crossover probability [%]	Mutation probability [%]	Best Score
50	70	6	0.7558
		10	0.7236
	90	6	0.7628
		10	0.7550

Some tests were run varying the number of iterations, but the figures below illustrate the differences between crossover probabilities of 70% and 90%, and mutation probabilities of 6% and 10%.

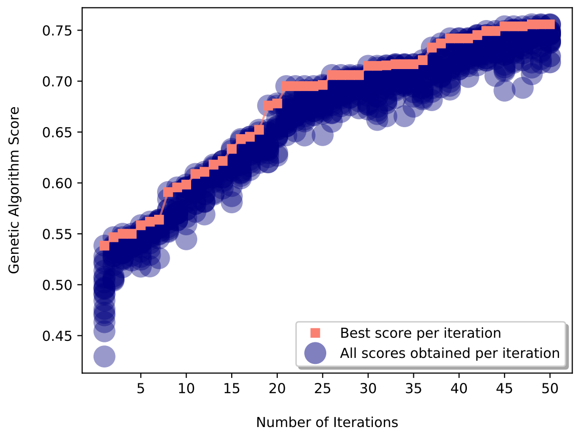


Figure B.1: Genetic algorithm evolution (70% crossover and 6% mutation probability)

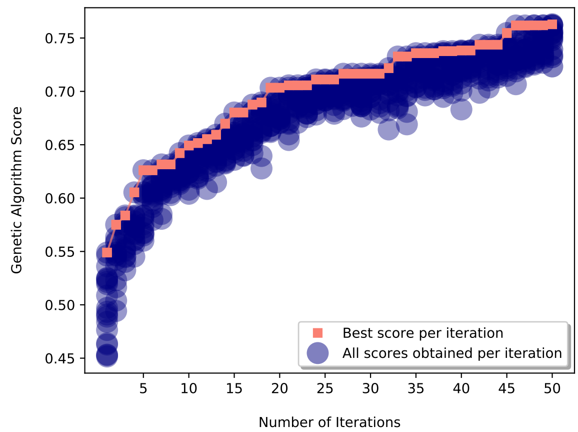


Figure B.2: Genetic algorithm evolution (90% crossover and 6% mutation probability)

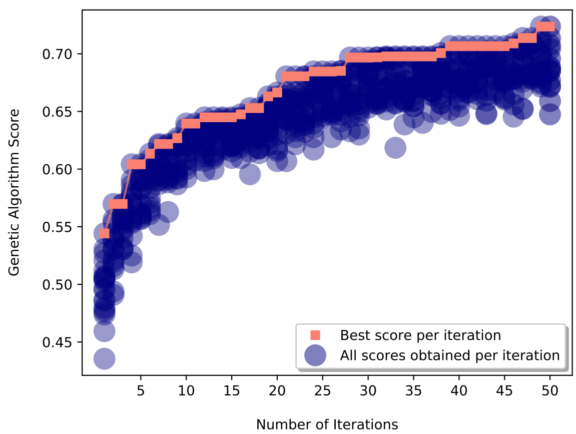


Figure B.3: Genetic algorithm evolution (70% crossover and 10% mutation probability)

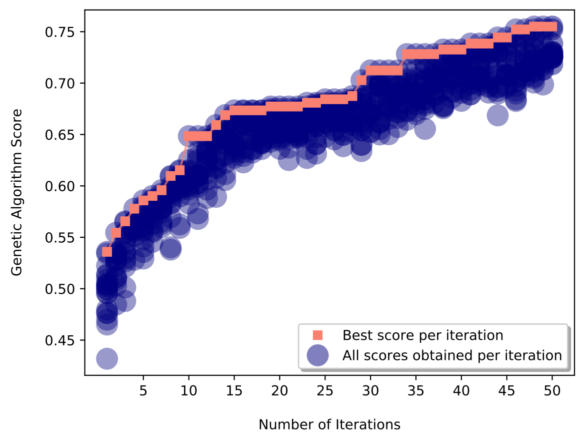


Figure B.4: Genetic algorithm evolution (90% crossover and 10% mutation probability)

From the genetic algorithm evolution, it's possible to visualize the difference between the algorithms' evolution with the different mutation probabilities and how it affects directly the exploratory character of the algorithm. The mutation brings diversity to the population which is crucial to improving the current population but it can be a risk since a mutation can also bring a downgrade. This has to be tuned considering other features such as elitism since this feature lowers the risk of wandering away from the desired goal. From the examples, the crossover probability does not have such an impact, possibly due to the lack of diversity in the initial population (only 10 individuals).

In addition, even though the results can be expected, some simulations were run for a higher iterations number - 100 and 200 - to show the evolution in the long run. From Figures B.5 and B.6 it is possible to comprehend the *exponential behavior* inherent to genetic algorithms. A simple explanation relies on initially the crossover having an important role in the initial diversity of the population but, in the long run, if no new individuals are entering the population, the diversity's source is mostly focused on the mutations in the population (a slower process).

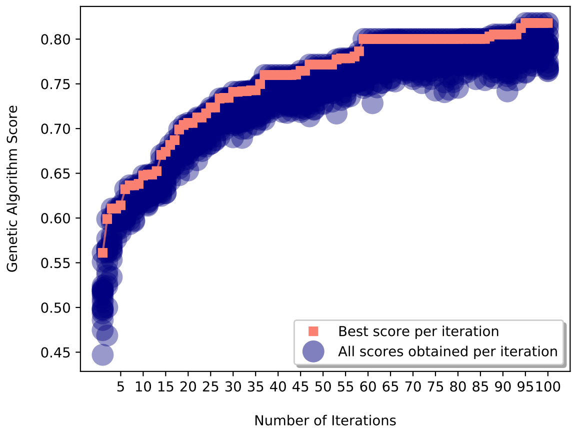


Figure B.5: Genetic algorithm evolution (90% crossover and 6% mutation probability) with 100 iterations

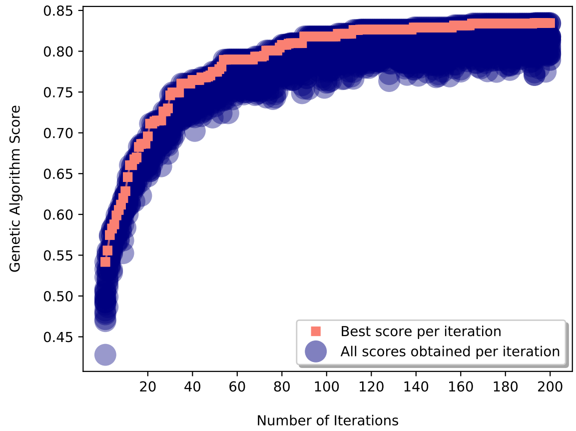


Figure B.6: Genetic algorithm evolution (90% crossover and 6% mutation probability) with 200 iterations

C

Conflict Detection Performance for Climb, Cruise and Descent

The conflict detection performance assessment is carried on including all the different flight phases. Thus, when looking at the evaluation parameters (TP, FP and FN) it can be questioned why the ratio of TP over the total is low compared to the ratio of FP over the total number of conflicts, and if this means the performance of a state-based conflict detection itself is intrinsically low. In order to investigate the previous hypotheses, the trajectories were classified with their flight phase and new results were obtained initially for cruise to investigate whether the issue was on the performance of the state-based method or if it was a consequence of considering climbing and descending trajectories.

An initial section covers the process to reach the new results in section C.1, which includes the flight phase identification (section C.1.1) and the process to obtain results making a distinction between flight phases (section C.1.2). Then, the difference between the results if highlighted in section C.2.

C.1. Implementation

C.1.1. Flight phase identification

The software used to simulate the air traffic scenarios includes a flight phase categorization which is based on performance. This categorization does not fit the purpose since the goal is to obtain one of the following geometries/geometries combination:

- Single geometry considers *Climb*, *Cruise* and *Descent*, considering there can be "steps" in the trajectories such as there is a constant period in the descent trajectory.

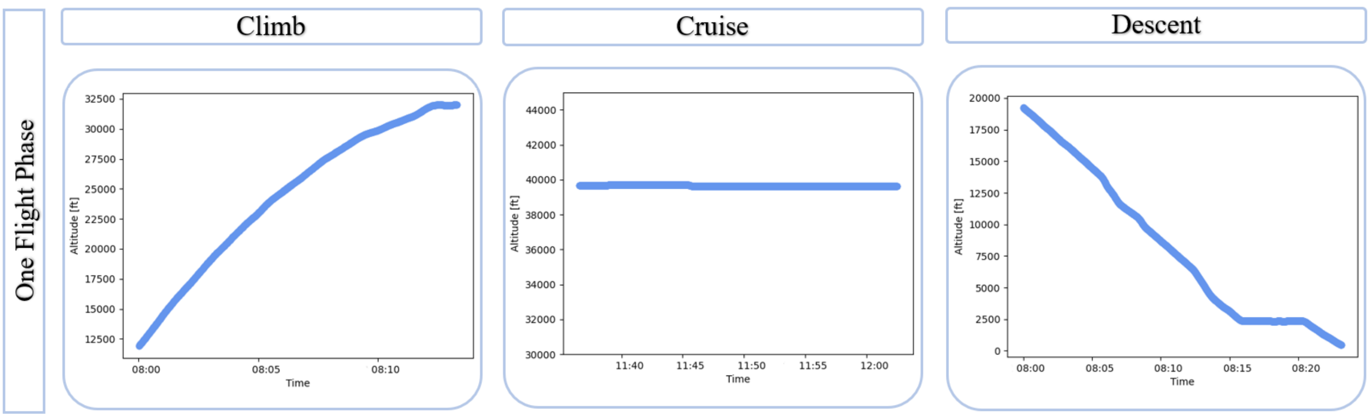


Figure C.1: Single flight phase options

- Two flight phases combine climb + cruise and cruise + descent, not allowing for climb + descent. Different geometries are evaluated and Figures C.2 and C.3 are some examples of some combinations. The examples illustrate the combination possibilities, highlighting the uncertain nature of the flight phase portion and proportion that can be in each trajectory.

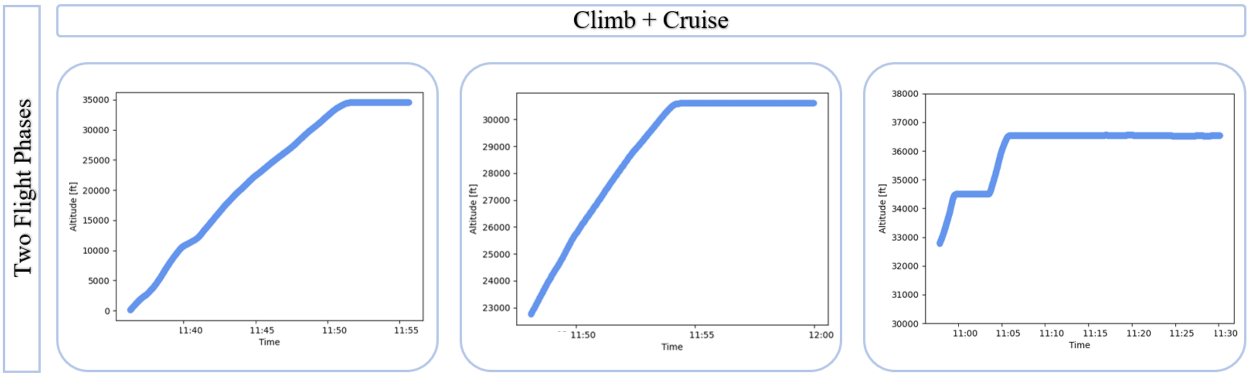


Figure C.2: Two flight phases - Climb + Cruise examples

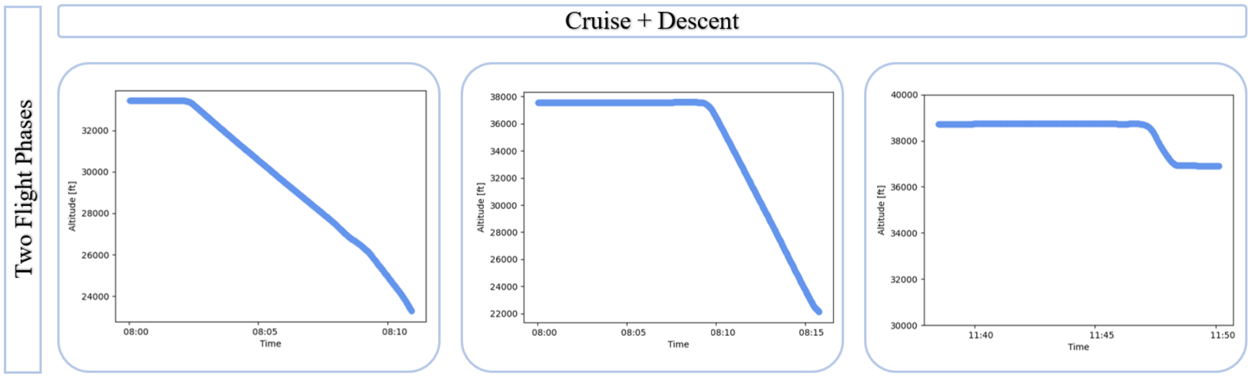


Figure C.3: Two flight phases - Cruise + Descent examples

- Three flight phases include all the categories and follows always an order of Climb - Cruise - Descent. Figure C.4 serves the same purpose as before, to illustrate a trajectory can include different proportions of the distinct flight phases.

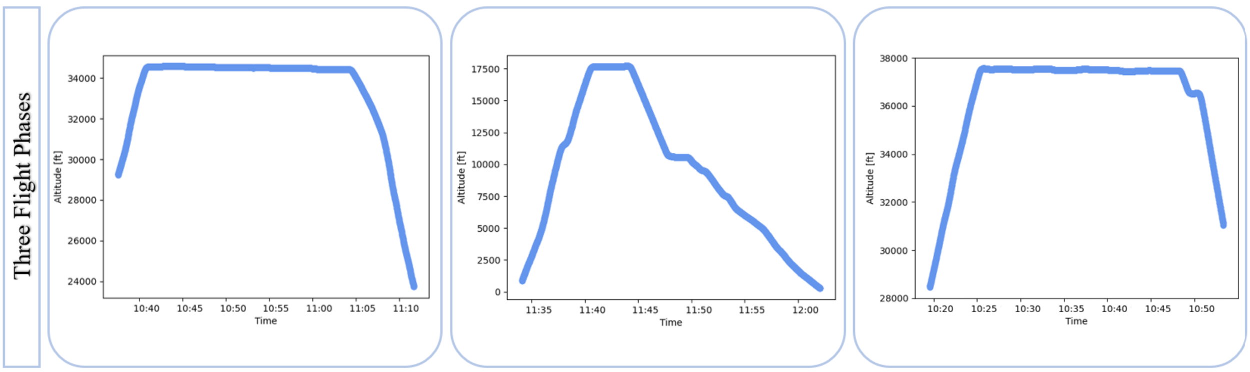


Figure C.4: Three flight phases examples

The process to compute the classification relied on geometric characteristics such as linear regressions' coefficients, coefficients from built linear functions with specific points, the error associated with these, among others. The process required some tuning and a try-and-error approach since the trajectories to be evaluated have all sorts of characteristics. Thus, several conditions were created for the identification of the different phases.

The process would initially rely on the observation of a linear regression of the whole trajectory extent and the error associated with it. Figure C.5 is a block diagram demonstrating the approach used. Then, the flight *BAW962H* will be used to exemplify the implementation of the algorithm.

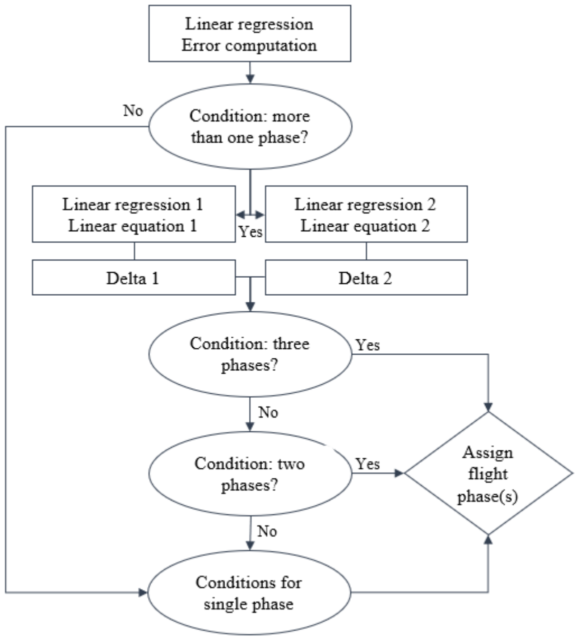


Figure C.5: Flight phase identification process

The example below illustrates the process of a three flight phases trajectory. Nevertheless, the conditions to the two flight phases and single flight phase are also explained. The first step as stated in the flight phase identification process, in Figure C.5, is the total trajectory linear regression computation and the respective error values. These are exemplified in Figure C.6 and Figure C.7, where the maximum value is only computed after the confirmation that is more than one phase and it separates one flight phase from the other(s).

The conditions to detect more than one phase focus on whether the error function crosses exactly twice the value zero or whether the maximum and minimum variation is higher than a certain threshold (in this case 2500ft). This is combined with the necessity to have the first and last error index lower than 1000ft (since the trajectory segments for more than one phase assimilate to a convex geometry and the error = original - linear regression).

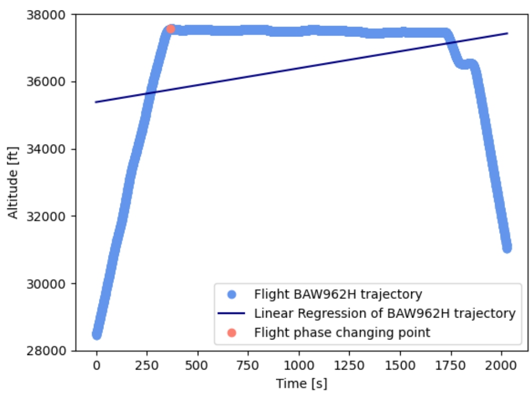


Figure C.6: *BAW962H* trajectory and linear regression

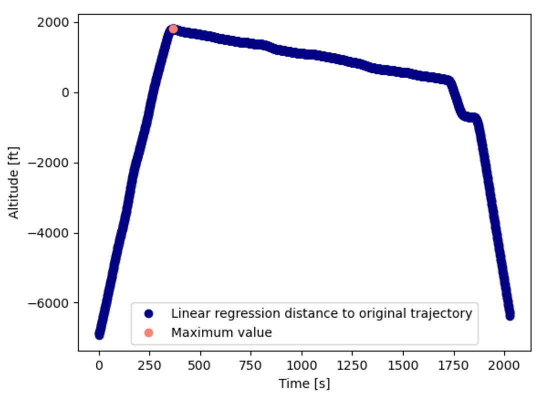


Figure C.7: Error between linear regression and original trajectory

In the flight phase identification process, the next step is to calculate the new linear regressions, considering the first phase changing point (the previous maximum value). In parallel, linear functions are computed using the first and last point of each segment, shown in Figure C.8. Then, a delta value for each segment is calculated from the linear coefficients. The conditions for the three flight phases evaluate which delta is higher (to identify the flight segment that potentially has more than one phase) and if it is higher than a minimum threshold (a percentage of the linear coefficients). If all conditions are verified, a new maximum is obtained for the error of the segment with the higher delta (Figure C.9) and the flight phases are attributed. Figure C.10 is an example of the output obtained.

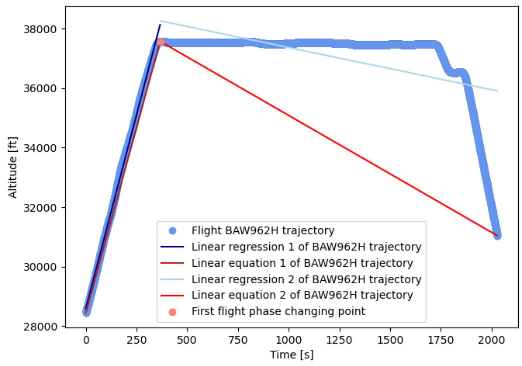


Figure C.8: *BAW962H* trajectory, linear regressions and linear equations

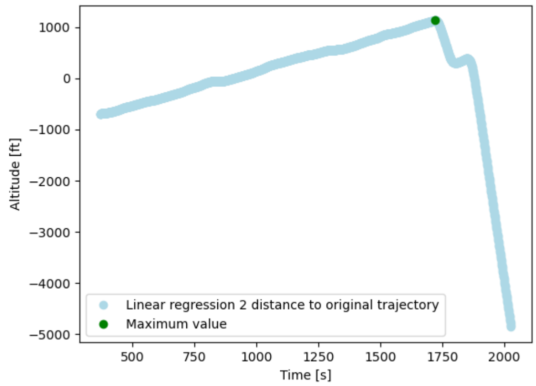


Figure C.9: Error between linear regression 2 and original trajectory

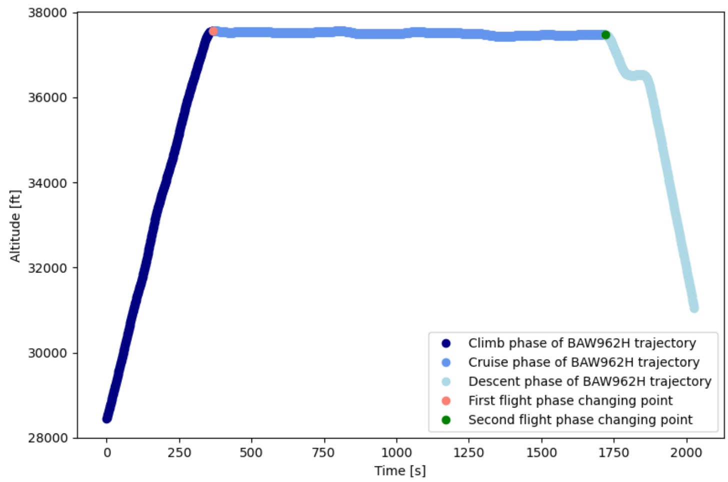


Figure C.10: *BAW962H* trajectory with flight phases assigned

The identification continues if the conditions for the three flight phases are not fulfilled. The conditions for the two flight phases focus on identifying which segment corresponds to the cruise phase (by limiting the slope coefficient between [-1.5,1.5]) or ensuring the delta correspondent is higher than a minimum threshold. If none of these comply, the single flight phase conditions are evaluated once again with the same criteria as before (the trajectories wrongly entered the if for more than one phase). The conditions consider cruise phase a coefficient between [-0.5,0.5], climb phase above, and descent below.

C.1.2. Conflict Detection Performance Calculation

The conflict detection performance calculator required the creation of an algorithm that reads the output data log file from BlueSky, detects the conflicts and categorizes them into the evaluation parameters (*true positives*, *false positives* and *false negatives*). The *Evaluation.py* file available in Github [1] includes the algorithm mentioned.

In order to investigate the results and assess the anomalies in the results, the flight phases were considered as explained. To include the flight phase information in the evaluation process, a data frame was created bringing together the information for every flight of when each flight phase starts (assuming a flight phase ends where the next one starts) and the elimination timestamp. Figure C.1 is a sample of a data frame, with flights with different flight phases' characteristics.

Time	Vertical rate [fpm]	Callsign	Flight Phase
28801	2648.00	*BEL7LY*	Climb
28912	0.00	*BEL7LY*	Climb
28801	-1203.20	*EWG8JM*	Descent
29035	0.00	*EWG8JM*	Descent
28801	1766.40	*RYR70JQ*	Climb
29126	476.77	*RYR70JQ*	Cruise
29673	0.00	*RYR70JQ*	Cruise
28801	1801.60	*EIN84T*	Climb
29524	0.00	*EIN84T*	Climb

Table C.1: Sample data from flight phase database

C.2. Results

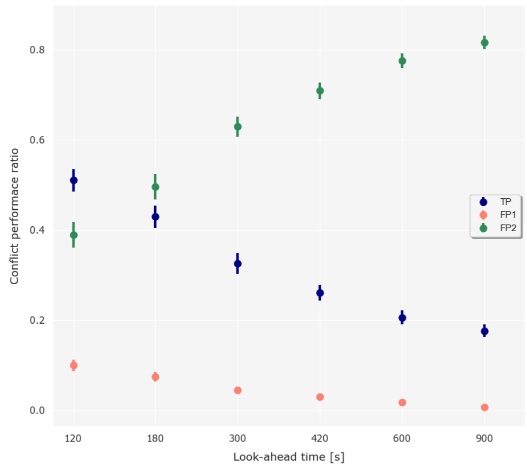
The results section for the flight phase impact on the conflict detection performance assessment covers all flight phases and both look-ahead time approaches. In addition, the correlation with traffic density and meteorological conditions bins is also explored.

Figure C.11 shows the results for all flight phases, for the *flexible approach*. The results include the performance analysis for each flight phase, to understand the impact each one has on the global performance. The conclusions in section 3 suggested the performance of climb and descent flight phases were considerably worse than the cruise phase, which is verified when looking at the plots. The extra analysis on this topic comes from comparing the performance for conflicts in climb phase with the performance for conflicts in descent phase and understanding their differences.

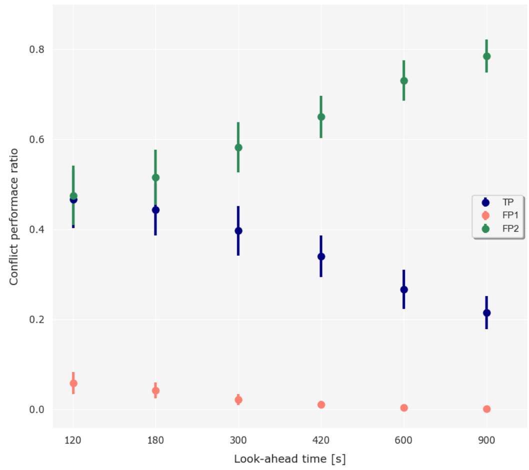
Climb and descent phases have several different characteristics, but only the ones identified as potential diverging agents will be mentioned. On the one hand, climbing trajectories follow overall higher vertical velocity values (absolute values) and more abrupt velocity changes than the descending phase, having steeper and more geometric progress. Therefore, there is a higher risk of conflict detection that turn out not to be an LoS with aircraft already in cruise at different flight levels. This would make climbing trajectories have more FP2 than descent trajectories. On the other hand, descent trajectories have very different characteristics throughout the trajectory (different descent angles and velocity trends) as opposed to climbing trajectories that usually follow approximately the same climb angle or have a step in between climbing segments. This difference could be in the origin of the higher number of FP1 for the descent trajectories, considering the conflicts do happen but the prediction is not as accurate as desired.

The evolution as a function of look-ahead time seems to follow a trend, having climb and descent with opposed evolutions. Climb's performance is more consistent for the lower look-ahead time values, while descent trajectories have a clearer evolution for lower look-ahead time values and stagnating progress for higher look-ahead times.

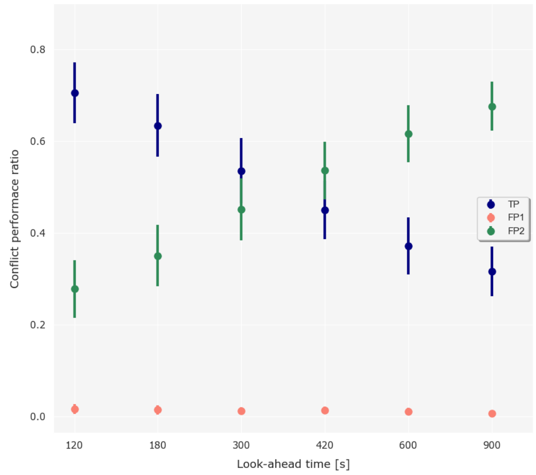
[1] `https://github.com/lunajuliao/CDperformance`



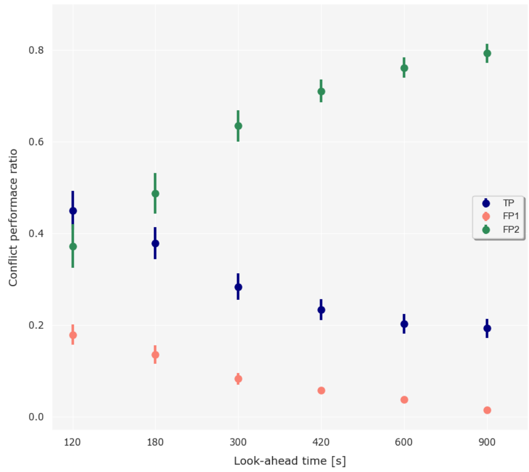
(a) Global Results



(b) Climb Results



(c) Cruise Results



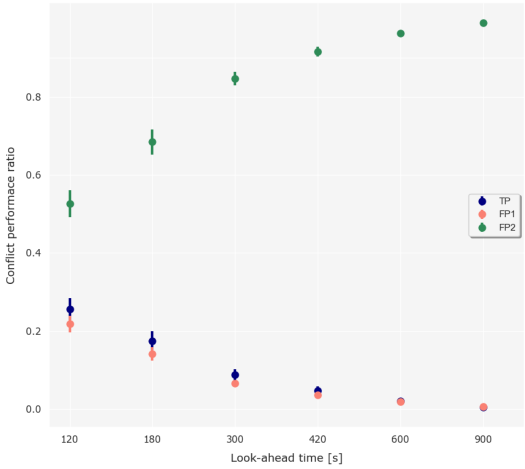
(d) Descent Results

Figure C.11: Flexible Approach for flight phase analysis

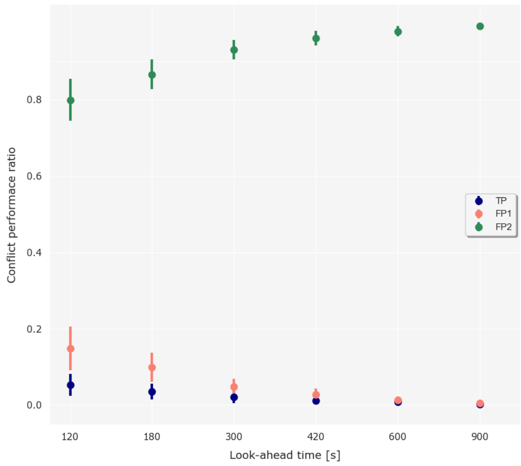
The *strict approach* in Figure C.12 emphasizes the analysis made before, specifically looking at the FP1 and FP2 difference in the climb and descent results. However, in this approach, the TP and FP1 relative positions are switched, having more FP1 than TP for these flight phases.

The previous discussion was on climb and descent trajectories' conflict detection performance differences, leaving cruise out of the overall analysis. For the traffic density and meteorological conditions bins, only those two phases will be considered.

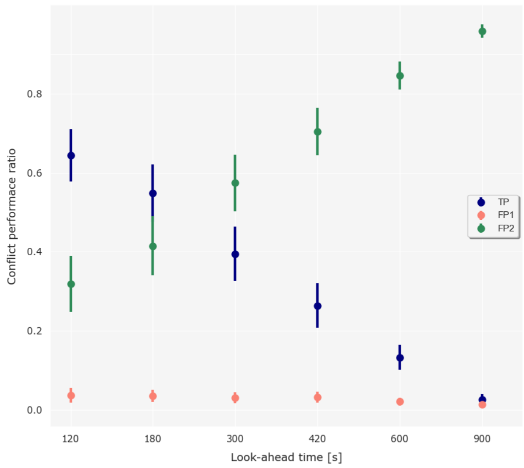
The results for climb and descent phases assessing the traffic density and meteorological conditions' impact do not provide clear trends from which is possible to withdraw conclusions. As before, for both approaches, the trend for the traffic density's impact is easier to identify than the meteorological conditions' effect. However, for both secondary variables, the values are not as distinctive: the average is very similar and the standard deviation is broader than what is allowed for concrete interpretations (the results overlay significantly).



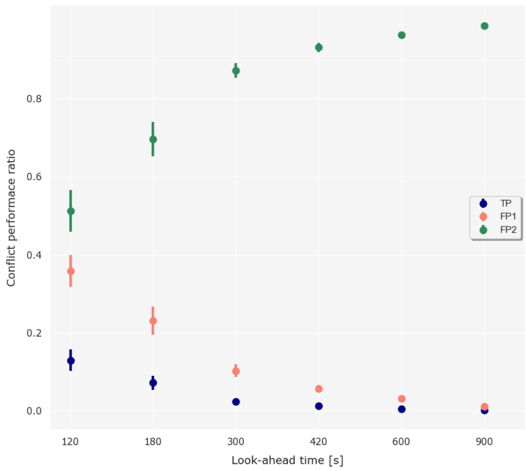
(a) Global Results



(b) Climb Results

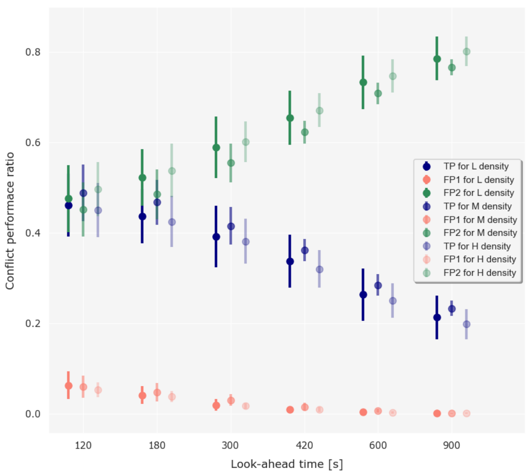


(c) Cruise Results

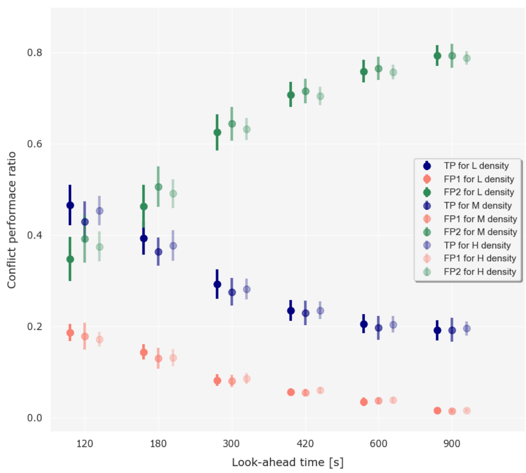


(d) Descent Results

Figure C.12: Strict Approach for flight phase analysis

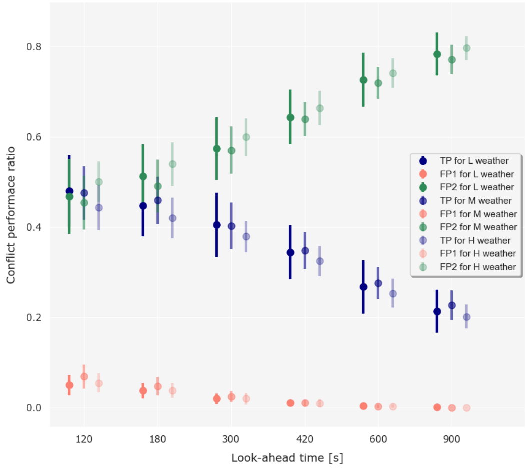


(a) Climb Results

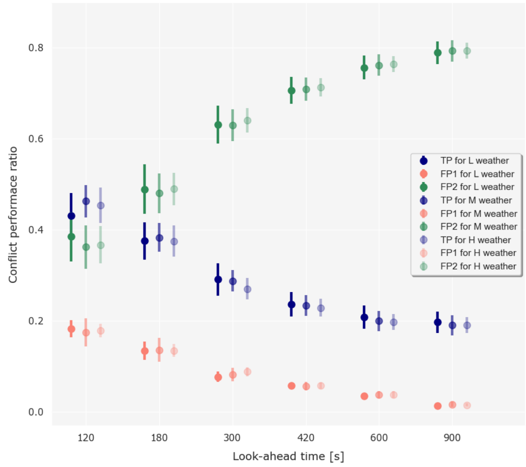


(b) Descent Results

Figure C.13: Flight phase and traffic density bin analysis for *flexible approach*

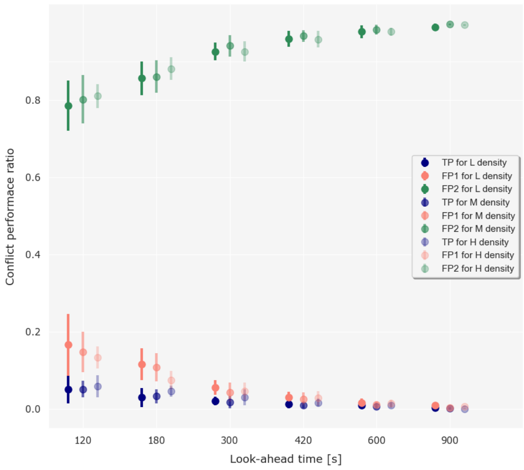


(a) Climb Results

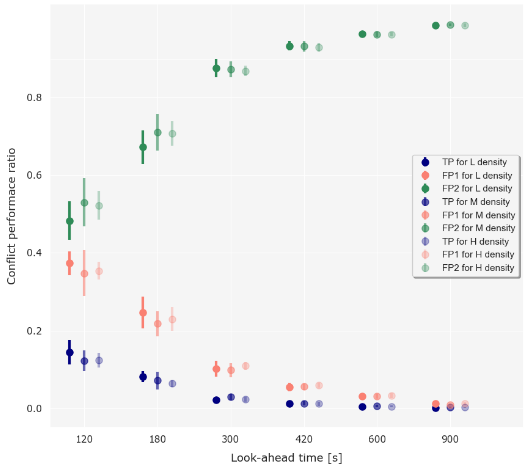


(b) Descent Results

Figure C.14: Flight phase and meteorological conditions bin analysis for *flexible approach*

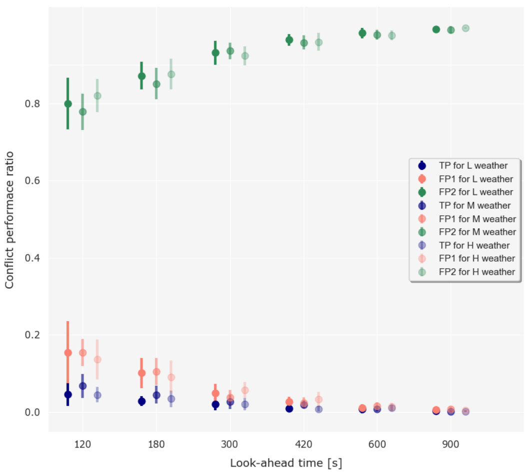


(a) Climb Results

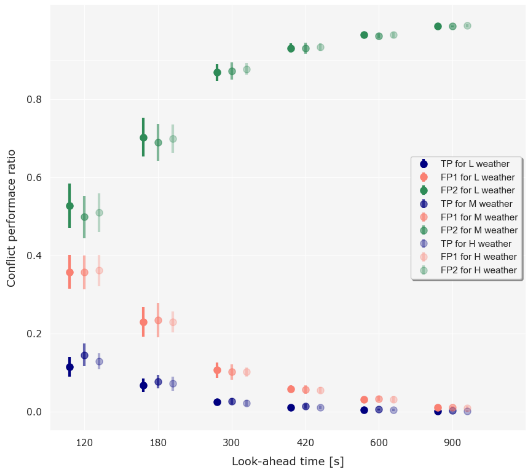


(b) Descent Results

Figure C.15: Flight phase and traffic density bin analysis for *strict approach*



(a) Climb Results



(b) Descent Results

Figure C.16: Flight phase and meteorological conditions bin analysis for *strict approach*

D
Validation & Extra Analysis

This section builds on the previous analysis and conclusions, focusing on the differences between the look-ahead time approaches and the scenario classification generation. A detailed look from the bin construction to the final scenario characteristics is included, to assess the validation of the traffic density and meteorological conditions bins.

D.1. Look-ahead time approaches

The results showed a considerable higher performance (TP vs. FN) for the *flexible approach* compared to the *strict approach*, mostly for higher look-ahead times. Figure D.1 illustrates why this happens. The FP concept common to both approaches in which a LoS is predicted but does not happen, FP2, has the exact same values for both approaches. The TP and FP are the parameters making the distinction between the approaches, thus, Figure D.1 illustrates the conflicts (from scenario one) considered TP for the different look-ahead times, for both approaches.

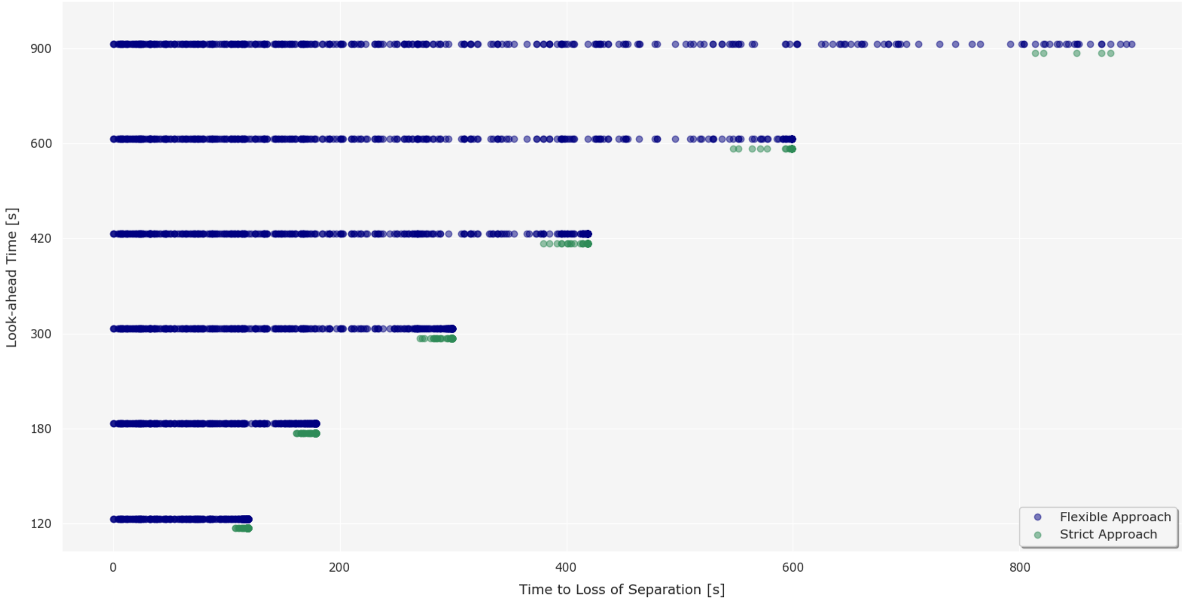


Figure D.1: Time to LoS for the different look-ahead times - *true positives* both approaches

The time to LoS gap for each look-ahead time for the *strict approach* is built based on the look-ahead time and buffer (10%), growing for higher look-ahead times. However, it is possible to observe for higher look-ahead times the TP density for that gap decreases. The comparison between this and the *flexible approach* distribution denotes the majority of the TP contribution is for time to LoS lower than 200s. Thus, the conflict detection performance for the *flexible approach*, mostly for higher look-ahead time values, can be misleading.

D.2. Scenario classification

The look-ahead time performance assessment considering traffic density and meteorological conditions bins presented some trends, generally consistent with the overall flight geometries. For particular flight phases, in some cases, some exceptions would occur. Hence, a detailed analysis of the traffic density influence was done and Figure D.2 and Figure D.3 are used as support for the findings.

Figure D.2 shows the total conflict number (TP + FP1) for the *flexible approach*, considering the bins distribution. A clear trend is visible, inside each meteorological conditions bin, for the traffic density bin influence since the conflict number increases with the correspondent increase in traffic density. This evolution is more substantial for the *Low* and *Medium* meteorological conditions compared to the *High* bin. This phenomenon could partially validate the meteorological conditions bins creation since it would be expected an increase in the conflict number for harsher meteorological conditions.

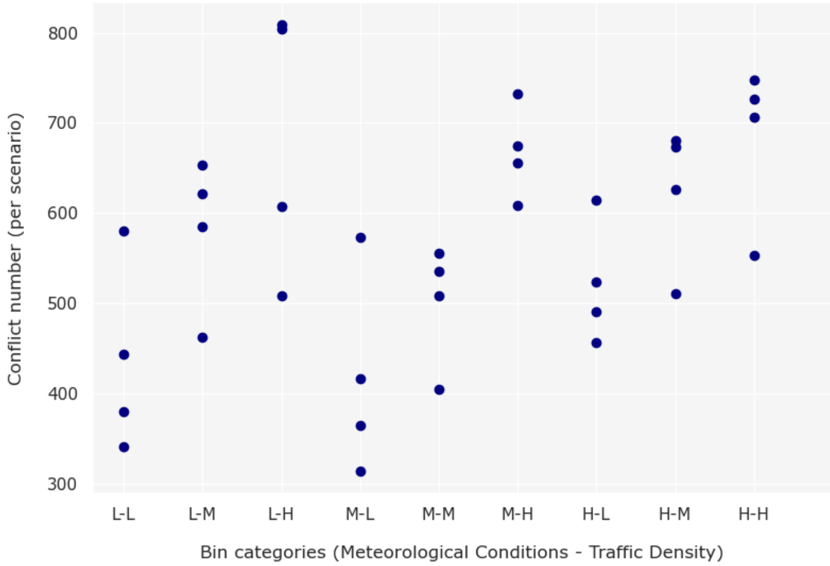


Figure D.2: Conflict number per scenario

Nevertheless, some extra analysis on the traffic density implementation was explored, to understand whether the conflict number distribution has some underlying motive. Therefore, the aircraft number for each scenario, organized per bins, is displayed in Figure D.3. The *Clustering results* are the values obtained after the *k-means* clustering divided by three since the values used were considered for the whole day and the scenarios considered in *Processed results* are 6-hour scenarios (the assumption of uniform traffic throughout the day can be introducing some bias in the whole process). For the *Clustering results* it is clear the non-overlap between the bins. However, the *Processed results* don't have the same characteristic, most likely due to the previous motive and the elimination of any general aviation and military aircraft. These last steps can interfere with each scenario very differently, resulting in a "biased" traffic density categorization. For future work on this topic, a larger set of scenarios could be picked and the traffic density could be reassessed after the basic processing.

The meteorological conditions classification can not be directly validated as the traffic density since it does not have any direct relation to the flight data. A potential justification for the unclear trends is the classification of the meteorological conditions considered scenarios for high bins that presented cold fronts in smaller geographical regions (e.g. cities), while the remainder of the geographical area presented good meteorological conditions, allowing for unaffected traffic, thus, with no different effect in conflict detection methods' performance.

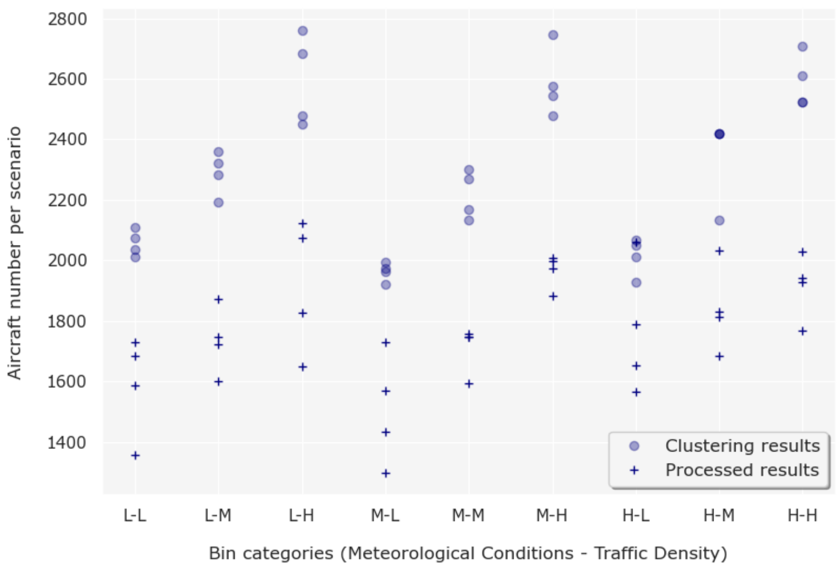


Figure D.3: Aircraft number per scenario

Bibliography

[1] ATAG. "*Revolutionising Air Traffic Management. Practical steps to accelerating airspace efficiency in your region* [pdf]". In: (2012).

[2] Joost Ellerbroek and Jacco M Hoekstra. "BlueSky ATC Simulator Project: an Open Data and Open Source Approach Three-Dimensional Airborne Separation Assistance Displays View project BlueSky-Open source ATM simulator View project BlueSky ATC Simulator Project: an Open Data and Open Source Approach". In: *Seventh International Conference for Research on Air Transport (ICRAT)* (2016).

[3] *European aviation facing serious capacity challenges now and in the future [online]*. `https://www.eurocontrol.int/press-release/european-aviation-facing-serious-capacity-challenges-now-and-future`. Accessed: 2021-05.

[4] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, 1992.

[5] *Machine Learning Mastery[online]*. `https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/`. Accessed: 2021-05.

[6] *Pypi (Genetic Algorithm Results) [online]*. `https://pypi.org/search/?q=genetical+algorithm&o=`. Accessed: 2022-02.