ELSA Air Traffic Simulator: an Empirically grounded Agent Based Model for the SESAR scenario

C. Bongiorno[1], S. Miccichè[1]
[1]Dipartimento di Fisica e Chimica, Università di Palermo
Viale delle Scienze, Ed. 18
I-90128 Palermo, Italy

M. Ducci[2], G. Gurtner[2]
[2]Deep Blue s.r.l.
Piazza Buenos Aires 20
00198 Roma, Italy.

Foreword - *This paper describes a project that is part of SESAR Workpackage E, which is addressing long-term and innovative research.*

***Abstract*—This paper describes the Air Traffic Simulator produced by the ELSA project. It is partially based on interacting agents taking actions during strategic and tactical phases: air companies, network manager, pilots, and controllers. The simulator is highly modular and each part can be used independently of the others. The code is open source, ready to use and available for the research community.**

Some results concerning the future organization of the European Airspace (free-routing) are presented, using the full capabilities of the model. We found that the implementation of free-routing could have a positive impact on the safety event occurrences that will be reduced in number although spread over a larger area. The controllers behaviour will therefore move to a situation where they have to perform a smaller number of operations dispersed over a larger portion of the airspace. We also show that the number of operations performed by a controller quadratically depends from the number of aircraft present in the considered airspace and that such quadratic scaling law is modified when the airspace is partitioned in air traffic sectors with capacity constraints.

I. Introduction

Even though the recent economic crisis has dumped the increase, it is foreseen that in the near future the European traffic will be too dense to be handled using the current procedures. SESAR has undertaken the difficult task of defining, exploring, testing, and implementing new solutions in order to tackle this issue [1][2][3][4]. Among them, the focus on the trajectories themselves has led to the concepts of free-routing and 4D trajectories[1]. With them, the airspace users will be able to plan an optimum trajectory right from the beginning and modify it at will during the flight. This will give a more "informative role" to the controller, as opposed to the full "directing" role he/she has today. These free-route operations first started in Maastricht airspace and are now extended to several countries in Europe [5]. More than ten ACCs had already implemented various steps of free route Operations. Some countries, like Spain and Hungary, are working to implement H24 Free Route Operations, while others such as Italy, Slovenia and Moldova are validating Night Free Route Operations. Moreover further expansion of Free Route Airspace Maastricht is planned before summer 2014. The challenge for the ATM is then to ensure a sufficient level of safety, which is cognitively difficult for controllers due to the increase of traffic complexity. To this end the WP-E ELSA project has developed an Air Traffic Simulator able to give insights about the wide-spread implementation of these new solutions and the impact on safety.

The Simulator is based on different independent modules, which, used together, give the opportunity to build a custom airspace, run a strategic phase on it, run a tactical phase, and post-process the results. Depending on the user's preference, the whole simulation can span the whole spectrum from fully data-driven to fully synthetic run. The former allows calibrating and validating the model, whereas the latter is used to generate stylized facts about a specific scenario. The Simulator features air companies, a network manager, pilots and ATCs, which are the relevant actors for the solutions we are studying. The simulator allows to model both the current and the future ATM scenario. For the latter the model includes the possibility of simulating some of the features foreseen by SESAR Step 1 Time Based Operations [6]. In particular the user can model:

- The implementation of free-routing by making requested trajectories from airlines progressively straighter across sectors and FIRs;
- The improved coordination, information sharing and trajectory prediction by simulating conflict-free planned trajectories;
- An extended controller look-ahead time up to 40 minutes.

The ELSA Air Traffic Simulator has been developed following a criterion of simplicity to allow users to fully control the model inputs. Other similar tools are available for research purposes such as CATS (Complete Air Traffic Simulator), FACET (Future ATMConcepts Evaluation Tool) [7][8][9] and AgentFly [10]. They feature advanced aircraft performance profiles, airspace models, weather data, and flight schedules for the testing of new ATM concepts. However none of them is publicly available and they are less prone to be easily modified

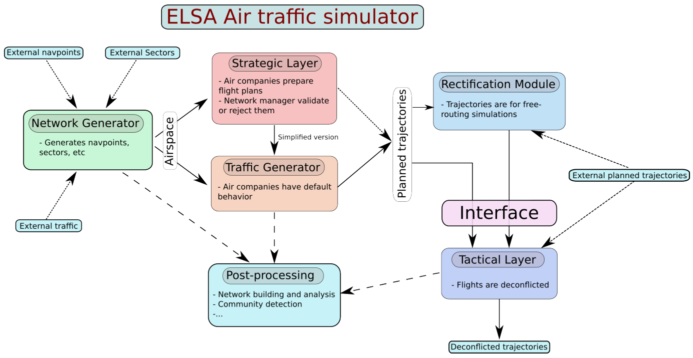

Fig. 1. Organization of the model.

directly by the end-users. The ELSA Air Traffic Simulator is instead intended as a flexible tool to test high level concepts of air traffic management in a quick and easy way to highlight general behaviors and to test the overall performance of the system. The aim of our tool is to provide the users with a cross-platform open-source simulator that can be used by different research groups to assess and compare their researches in the same software environment. Although not included in the present version of the simulator, more advanced features (e.g. aircraft performance profiles, weather data, etc.) can be integrated with limited effort according to the specific user's needs.

The paper is organized as follows. Section II is describing the Simulator, the organization of the code and some explanations of the relevant parts of the model. Section III describes briefly where the code has been released and the type of license we used. Section IV is dedicated to the calibration of the model, whereas section V shows some results when we implement the free-routing within the simulator. We finally present some conclusions in section VI.

II. The Model

The ELSA Air Traffic Simulator is composed of several fairly independent modules:

- A **network generator**, including navpoints and sectors.
- A full **strategic layer** and a simplified **traffic generator**, used to test different traffic situations,
- A **rectification module**, used to straighten up trajectories when simulating free-routing,
- A **tactical layer**, with a conflict resolution engine, simulating a tunable, imperfect super-controller.
- A **post-processing** module, including standard metrics computation and a simple graphic interface to see isolated run.

A schematic representation of the code is displayed in Fig. 1, where the interconnections between the different modules of the model are clearly shown. Moreover, some of the modules of the present model have already been described in Ref. [13] and Ref. [15].

The model uses a description of the airspace in terms of navigation points (which form a network on which the flights are travelling) and sectors. The latter defines in which area the controller can actually interact with the flights.

The core engine of the model is the resolution of conflicts, given an airspace and some flights travelling through it. The controller can have different strategies (horizontal deviation, vertical deviations, give a direct) depending on the environment and its own limited – and tunable – look-ahead time. The controller is perfect in the sense that it avoids all conflicts but sometimes makes suboptimal decisions due to its limited forecast capacities. "Shocks" can be used to perturb the trajectories. Indeed, some parts of the airspace can be randomly shut down for a given time, simulating weather events or military exercises. The spatial and temporal distributions of shocks can be fully controlled.

The model also includes a possibility of building some airspace, ranging from full manual definitions of navpoints and sectors to automatic generation of airspace. Real airspaces can also be used easily, possibly integrating deviations from reality controlled by the user (e.g. number of navpoints). The airspace can be composed of several sectors and is in three dimensions.

The model gives the possibility of generating flight trajectories in a controlled way. In particular, there exists the possibility of slowly tending to business trajectories – i.e. straighter and straighter trajectories, thus testing the possibility of continuous integration of the free-route scenario on real airspace. The integration can be heterogeneous, with some sectors keeping a fixed grid of navpoints whereas others have already moved to business trajectories.

A. *Network generator*

The simulator comes with a network generator module which is very versatile. In particular, it allows:

- To generate the spatial distribution of navigation points or use external data,
- To compute the navigation points network edges with a triangulation[1] or use external data,
- To generate sectors at random, using a Voronoi tessellation [12] for the boundaries or use external data,

[1]We use the Delaunay triangulation for its properties [11].

- To compute time of travels between edges of navigation points or use external data.

Hence the user can fully specify the network and the sectors or use the module in a semi-automated way. It is also possible to build a network based on traffic data.

B. Flight Plan Generator

The trajectories generator can be used to generate synthetic traffic on a given network of navigation points and sectors. It allows to create traffic in a set of sectors given some airports and/or entry/exit points in a realistic way, making sure no sector is overloaded. The user can specify in particular:

- A total number of flights,
- a distribution of flights per pair of entry/exit points,
- some capacities for the sectors,
- a distribution of flight levels occupancy,
- the navigation point network to be used.

The flight plan generator is a simplified version of a bigger model, the "strategic layer", which is fully available in the same repository. The strategic layer is full Agent-Based Model where different entities (airlines and network manager) are collaborating or competing for the same resources (time slots and trajectories). It is described more in detail in [14]. Note that the strategic layer and its simplified counter-part are designed to generate trajectories with a coarse level of description, suitable to study high level phenomena. In particular, the trajectories are kinematic and do not take into account winds, weight, etc.

C. Rectification module

The rectification module was introduced to study the transition between the current scenario and the future free-route scenario in a controlled way. The module requires as input a generic M1 file, i.e. a set of planned trajectories in the current scenario, and produces as output another M1 file where trajectories have larger target value of Efficiency. This metric is defined as the ratio between the actual length of the trajectory and the shortest path between origin and destination [18]. At each step the algorithm evaluates the current Efficiency and if it is less than the target Efficiency it substitutes a point of a route randomly selected with the medium point between the previous navigation point and the successive.

D. Conflict Detection and Conflict Resolution modules

This module is the core of the tactical layer of our simulator. In this module the relevant agents are aircraft present in a certain airspace and the air traffic controllers that manage their trajectories. The model we present hereafter is quite simple and can be considered as a zero-intelligence model where we do not have learning and the agents interact in a mechanistic way.

This module is used to check whether any safety event occurs and to solve it [18]. It essentially works along the same lines than the analogous module already presented in Ref. [13]. The issue of Conflict Detection and Resolution has been extensively dealt with in the literature. Given the present length constraints, we can refer the reader to a short review, available at `http://www.complexworld.eu/agent-based-models-take-off/`, where the main ideas and methodologies are briefly discussed, also with reference to the WP-E research projects.

In order to check for collision between an i-aircraft and all the other $N_f - 1$ ones, the Conflict Detection module performs a subdivision of the time-step into N elementary time increments δt and computes the position of each aircraft at each elementary time increment. We thus have an array of positions for each aircraft present in the considered airspace. Then the module compares the position-array of the i-aircraft with the position-arrays of the other $N_f - 1$ aircraft by calculating the distance between any two aircraft at each time increment. If at least one value is below the minimum separation distance of 5 nautical miles then a conflict is detected.

The simplest version of the Conflict Resolution module is based on two strategies: rerouting and flight level change. In order to solve the conflicts the module first tries to reroute the aircraft then to change its flight level. To reroute the aircraft the Simulator generates a set of random temporary navigation points around the location of the possible conflict within a range of 100 Km. Then it tries to send the aircraft towards one of these temporary navigation points selected with the criterion of (i) minimizing the total path length and (ii) with the constraint that the angle between planned and deviated trajectory must be smaller than a threshold value selected by the user. If no solution is found by performing a re-routing then the module tries to change the flight level. It first tries to send the aircraft 2 FL up then it tries 2 FL down.

E. Pre-tactical de-conflicting module

The task of the pre-tactical de-conflicting module is to generate conflict-free planned trajectories starting from real or surrogate planned trajectories. The need for such a module is due to the fact that one of the features foreseen by SESAR will be a better planning of the trajectories such that they may be already conflict-free [6].

As such, since we are still at the planning level and no issue regarding the flight conditions is taken in consideration, differently from the module of II-D, that modifies the sequence of navigation points/flight levels, this module only acts on the departure time of the aircraft.

In this case we maintain the sub-module that performs the Conflict Detection. Such module is now associated with a Shift-In-Time module that simply randomly changes the departure time of an aircraft experiencing a separation minima infringement in its journey. Another big difference with the module of II-D is the time-step which is now fixed to 24 hours, because we want to perform such de-conflicting at a daily level. Unfortunately this requirement implies the use of a huge amount of memory and therefore the module can be computationally time-consuming.

The Conflict detection module checks for any possible conflict according to a list. If it detects a conflict it tries to shift in time the departure of the aircraft of an amount of time within the range [-5 min, 5 min]. It tries this procedure until the flight trajectory is de-conflicted, for a maximum number of 100 iterations. If at the end of the 100 iterations the aircraft is still experiencing a loss of separation it tries to shift in time the departure of this aircraft of an amount of time within the range [-10 min, 10 min], then within the range [-15 min, 15min] and finally in the range [-20 min, 20 min], if any solution is found the module starts again with another ordering of the list.

In most practical cases at the end of this process all planned trajectories will be conflict-free. Note that the pre-tactical de-conflicting model does not take into account any kind of uncertainty. It is simply designed to avoid the major part of the conflicts by solving those which would occur if all flights would go as planned.

F. Shocks module

This module is used to model the presence of portions of airspace that can not be crossed by any aircraft. Generically we call them shocked areas. They might be military areas closed to civil air traffic as well as areas with strong weather events that make them inaccessible for the aircraft.

The shocks are modeled as circles of center C_S and fixed radius R_S and located at a flight level drawn from a random uniform distribution in the range $[FL_{min}, FL_{max}]$. Each shocks vertically extends over one separation level, i.e. 1000 feet. Each shock has a duration D drawn from a random uniform distribution in the range $[1, D_S N \delta t_r]$. The area within these circles is inaccessible for all aircraft. In the model the number of shocks follows a Poisson distribution with mean S_m. The position of the shocks is drawn from a list of points provided by the user.

At the beginning of each time-step the controller cannot forecast the shocks. This means that he/she looks at the current position of the shocks and he/she operates assuming that the shocks are fixed along the time-step Δt even if they could disappear within the time horizon.

The module is integrated in the simulator in such a way that any user, having its own list of shocked areas, can use this list as an input to the model. The only requirement is that all the parameters characterizing the shocks, i.e. C_S, R_S, $[FL_{min}, FL_{max}]$ and duration D, are explicitly given.

G. Directs module

This module is used to model the fact that controllers might issue directs in order to shorten some aircraft trajectory.

A direct is made by removing one or more navigation points of the planned flight-plan after the first navigation point present in the current time-step. The module first evaluates how many navigation points can be removed with the constraint that the flight has to come back on the original route within a time interval equal to $T = 2\Delta t$, without infringing the sectors capacities of the nearby sectors. Moreover, the direct is not given either if the absolute difference between the length of the planned trajectory and the trajectory modified with the introduction of the direct is smaller than L_s and if issuing the direct generates a safety event within the controller's look-ahead.

H. Multi-sector features

We have also implemented in the model the possibility to have the airspace, typically an ACC, divided into different air traffic sectors. The model parameters that are used to model the way different controllers manage the trajectories are the same. This is equivalent to assuming that there exists a unique super-controller.

However, some genuine multi-agent features were also included. In fact, with the aim of modeling a simple coordination between sectors we imposed the following simple rule: sectors that reach their maximum capacity do not issue any directs and do not accept any direct from other sectors.

Another feature, implemented in the model but not considered in the results presented hereafter, is the tunable local knowledge of the airspace as measured by the controller look-ahead. In fact, a controller with a large look-ahead should have a better vision of the system also taking into account the air traffic of neighboring sectors. This should give him the possibility of issuing directs and conflict resolution strategies well optimized over a larger portion of airspace. However, the look-ahead of 10 min used in the results presented hereafter, should be already large enough to take into account most of this effect, given the fact that usually, in normal operational conditions, executive controllers know the flight trajectories up to five minutes before (after) the aircraft enters (leaves) their sector.

III. Code release

The Simulator is written in Python [16] and C [17], but only a limited knowledge of Python and C is required in order to use it. It is released under the General Public License version 3, i.e. it is open-source. In particular, it is freely downloadable on Github at the address `https://github.com/ELSA-project/ELSA-ABM`. The community is welcome to use it, modify it, ask for clarifications and report bugs, using the tools available on GitHub or contacting directly the authors. The code has currently been fully tested under Linux, but should work also with MACOS with minimal effort.

The interested reader will find on the above mentioned repository a full description of how the different modules have effectively been implemented in C and Python. These details will not be described here because we thought it was more important to show the results that can be achieved by using the model rather than focusing on technical details which would be more appropriate for an IT community.

IV. Calibration

The parameters entering the different modules described above are summarized in Table I. In the third column we give a short description of the parameters and in the fourth column we introduce a classification of the parameters in terms of the three categories described below:

- FP - free parameter, to be chosen according to the type of experiments one wants to perform.
- CD - parameter that needs to be calibrated from data.
- CV - parameter that needs to be calibrated according to the validation activities performed with ATM experts and ATCOs.

The parameters that need to be calibrated from data are really a few. However, there are many parameters (CV category) that are related to the behavior of controllers. In principle, these are parameters that could be inferred from data through some sophisticated data mining. However, we believe that these are the typical parameters that should be selected by consulting ATM experts and ATCOs. On the other hand these are the parameters that one should change at will when performing scenario simulations to test how changing a certain feature will affect the ATM system.

TABLE I
MODEL PARAMETERS.

ID	Param.	Description	Type
1	Δt	Length of the time-step. This is also related to the controller's look-ahead.	FP
2	δt	Length of the elementary time-intervals.	FP
3	t_r t_r	Fraction of Δt by which we move the overlapping time-steps.	FP
4	σ_v	Range of the noise introduced in the estimation of the aircraft velocity	CV
5	D_{max}	Radius of the circle centered in B where we look for temporary navigation points potentially relevant for for performing a re-routing.	FP
6	α_M	Maximum angle of deviation between planned and modified trajectory.	CV
7	T_{max}	Maximal temporal distance between the navigation point B and navigation point E that identify when a deviated portion of flight trajectory starts and ends.	FP/CV
8	p_d	Probability to try to issue a direct.	CD/CV
9	L_s	Sensitivity threshold for issuing a direct.	FP/CV
10	C_S	Center of each shock.	FP/CD
11	S_m S_m	Average number of shocks per time-step per flight-level.	FP/CD
12	D_S	Temporal duration of each shock.	FP/CD
13	R_S	Radius of each shock.	FP/CD
14	FL_{min}, FL_{max}	Minimum/maximum flight level where shocks are generated.	FP/CD

In addition to these parameters, the flight plan generator requires the specification of the distributions indicated in Table II. These are distributions that can be easily obtained from real data.

TABLE II
MODEL INPUTS.

ID	Param.	Description
1	v	distribution of the aircraft velocity
2	FL	distribution of the flight levels occupancy
3	xx	distribution of flights between origin-destination pairs
4	ntw	real navigation point network

As a result, the simulator described above can provide numerical simulations where the heterogeneity of the different flights can be fully exploited. For example, by considering the velocity, flight level occupancy and origin destination distributions we can fully take into account the heterogeneity of the flight trajectories. On the other hand, the behavioural parameters mentioned above might help in assessing the ATCOs heterogeneity of the behaviours.

V. RESULTS

We will present in this section some results showing how the simulator briefly sketched above can be used to perform policy experiments relative to the SESAR scenario. In section V-A and section V-B we will show how, in particular, in the free-route scenario we can expect that the controllers will have to perform a smaller number of operations although they will be more dispersed over a larger portion of the airspace. These results will be obtained in the idealized case where sectors' capacity do not play any role. Moreover we applied the pre-tactical deconflict module in order to have conflict-free routes, then we perturbed the system applying a random uniform delay between [-5m,5m] at the departures of the aricraft. Section V-A will also show that in this idealized case, the results are quite robust with respect to the number of aircraft present in the considered airspace. In fact, we can observe a scaling law showing that the number of operations performed by the controllers quadratically depends on the number of aircraft. In section V-C we show that this scaling law might be modified when the airspace is partitioned in sectors for which capacity constraints must be fulfilled. In this case the scaling law is still valid, although the scaling exponent is larger than in the case when no capacity constraints are present.

A. *Efficiences to SESAR*

In Fig. 2 we show the average number of conflicts detected in the LIRR ACC, for different values of efficiency (horizontal axis) and for different values of the aircraft present in the ACC (different lines in the plot). The simulations were performed by switching-off the module that implement the possibility of issuing directs. Each of the shown curves has been normalized with N_f^2, i.e. with the maximum possible number of conflicts in an environment with N_f aircraft. The average number of conflicts is here measured as the average number of actions that the controller has to perform in order to avoid the conflicts detected by the module of section II-D. Indeed, the controller of the model is taking exactly one action per conflict detected. Therefore, these measures are performed on the actual flight trajectories generated by our model. The figure shows two interesting features: on one side we have that all curves seem to collapse in a single curve when the number of conflicts is rescaled with N_f^2. The second interesting feature is that the number of detected conflicts diminishes as long as efficiency increases, thus indicating that in the free-route scenario we should observe less conflicts and therefore a smaller workload for controllers.

In addition, we have also devised a simple procedure to compute what is the expected number of possible safety events (PSE) i.e. separation minima infringements, we should expect. In this way, we can assess whether the results of Fig. 2

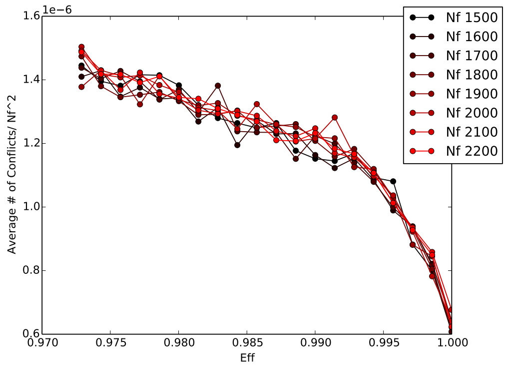

Fig. 2. Average number of conflicts detected in the actual flight trajectories of the LIRR ACC, for different values of efficiency (horizontal axis) and for different number of the aircraft present in the ACC (different lines in the plot). Each of the shown curves has been normalized with N_f^2 that represents the maximum possible number of conflicts in an environment with N_f aircraft.

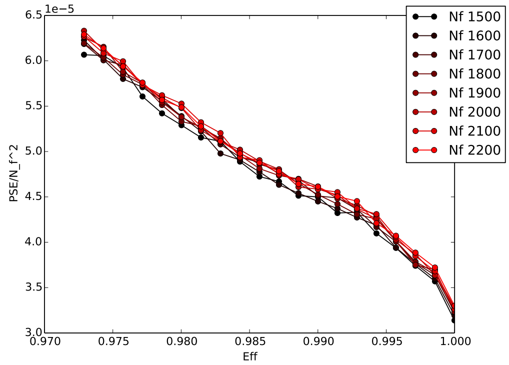

Fig. 3. Average number of possible safety events (PSE) detected in the planned flight trajectories of the LIRR ACC, for different values of efficiency (horizontal axis) and for different number of the aircraft present in the ACC (different lines in the plot). Each of the shown curves has been normalized with N_f^2 that represents the maximum possible number of conflicts in an environment with N_f aircraft.

are realistic or not. We start from the planned de-conflicted trajectories and implement the following procedure:

- we perform a very fine spatial sampling of all flight trajectories. Sample points are distant 1 meter one from the other.
- starting from the original flight plans, we associate to each of these sampled points a timestamp. This is done by assuming that between two navigation points the velocity of the aircraft is constant.
- we select those sampled points $P_i^{(f_1)}$ in the f_1-th flight trajectory and $P_j^{(f_2)}$ in the f_2-th flight trajectory such that the Euclidean distance $d(P_i^{(f_1)}, P_j^{(f_2)})$ between the two points is smaller than the separation minima $d_{thresh} = 5$ NM.
- we further select those points such that the times $t_i^{(f_1)}$ at which the f_1-th aircraft crosses $P_i^{(f_1)}$ and $t_j^{(f_2)}$ at which the f_2-th aircraft crosses $P_j^{(f_2)}$ are below a certain time threshold T_{thresh}.

By using such procedure we are able to show what are the points of the ACC that are likely to attract the controller attention as a source of possible safety events. Of course, the PSEs thus defined are strictly dependent on the T_{thresh} considered. In Fig. 3 we show the PSEs detected in the LIRR ACC, for different values of efficiency (horizontal axis) and for different number of aircraft present in the ACC (different lines in the plot). Each of the shown curves has been normalized with N_f^2, i.e. with the maximum possible number of conflicts in an environment with N_f aircraft. In the figure we show the results for $T_{thresh} = 5.0$ min although we performed such analyses for different values of T_{thresh}. Also in this case, the figure shows two interesting features: on one side we have that all curves seem to collapse in a single curve when the number of conflicts is rescaled with N_f^2 and the number of detected conflict diminishes as long as efficiency increases, thus indicating that in the free-route scenario we should expect less conflicts and therefore a smaller workload for controllers.

Note that figure 2 shows the *actual* conflict detected in the model, taking into account delays at departure and previous conflicts, whereas figure 3 shows the *possible* conflicts only based on the geometry of the trajectories.

In Fig. 4 we show a scatter-plot between the normalized PSEs detected from the planned trajectories with $T_{thresh} = 5.0$ min (horizontal axis) and the normalized number of conflicts detected from the actual trajectories (vertical axis) for different values of efficiency. The figure shows the existence of two different regimes. For values of efficiency close to unity the curve can be fitted with a linear relationship whose slope is of the order of 0.05, while for lower values of efficiency we have we have a linear relationship whose slope is of the order of 0.01. In any case, the fact that the curve is steeper for high values of efficiency indicates that a small variation in the PSEs translates into a larger variation of the number of detected conflicts, thus indicating that the free-route scenario might reveal to be less flexible to accommodate variation in the planning of the trajectories.

B. Heterogeneity

In Fig. 5 we show a density map of the PSEs detected when considering three different values of efficiency and $T_{thresh} = 5.0$ min. In the top panel we show the PSEs detected starting from the real planned trajectories, which corresponds to an efficiency value of $E = 0.9729$. In the bottom panel we show the PSEs detected starting from the planned trajectories corresponding to the free-route scenario, i.e. with an efficiency value of $E = 0.99999$. Please notice that in order to enhance readability, the number of PSEs in the two panels have been normalized. In fact, we first take the logarithm of the number of PSEs and then we normalize by dividing all logarithms by the maximum one. Therefore, the comparison between the two panels can only be done taking into consideration the spreading of the PSEs and not their values.

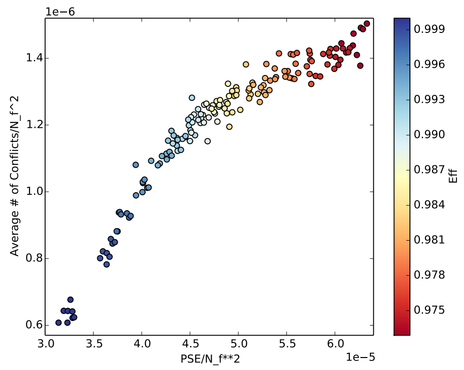

Fig. 4. Scatter Plot of the average number of conflicts detected in the actual flight trajectories versus the average number of possible safety events (PSE) of the LIRR ACC. Different points represent different values of efficiency and different values of the aircraft present in the ACC.

As expected, as long as efficiency increases the possible conflicts are more spread all over the ACC, rather than being concentrated in specific regions. This might explain why the number of detected conflicts diminishes when the efficiency increases. This also implies that the controller activity in the free-route scenario will change, moving from a situation where he/she has to give attention to an high number of conflicts concentrated in specific points to a situation where he/she will have to manage less conflicts spread over a much larger portion of the airspace.

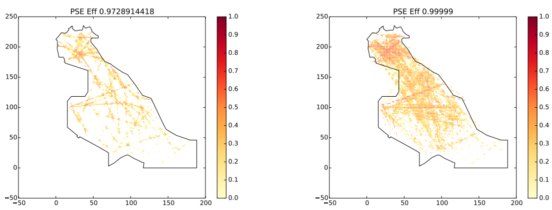

Fig. 5. Density map of the PSEs detected when considering three different values of efficiency and $T_{thresh} = 5.0$ min in the LIRR ACC. In the top panel we show the PSEs detected starting from the real planned trajectories, i.e. of $E = 0.9729$. In the bottom panel we show the PSEs detected starting from the planned trajectories corresponding to the free-route scenario, i.e. with an efficiency value of $E = 0.9999$. To enhance readability, we first take the logarithm of the number of PSEs and then we normalize by dividing all logarithms by the maximum one.

C. Scaling laws in a multi sector environment

The previous results rely on the fact that the trajectories are unconstrained, i.e. that they cannot be rejected because of some capacity constraints. However in reality, the number of flights per sector is limited. This has the direct consequence that for the same number of flights, the number of conflicts should be smaller in case of binding capacities, which is exactly the reason why they do exist. An indirect consequence could be the modification of the scaling observed in figures 2 and 3. Indeed, the very structure of the sectors should be designed to break the hard problem of solving simultaneously N_f^2 potential conflicts.

In order to study this point, we added a structure of sectors to the LIRR ACC. To keep the simulations simple, we took the sector tiling which exists at high altitude (FL350) and used it for all altitudes. In other word, the sectors are infinitely high. We then ran some simulations by generating synthetic planned trajectories, keeping the number of flights under the capacity within each sector.

The values of the capacities were partially inferred from data. With some real data of LIRR, we tracked the maximum number of flights crossing each sector within an hour and set this as the capacity. Fixing the total number of flights, we then decreased uniformly the capacities of all sectors and tracked the number of conflicts. In order to make a comparison with an unconstrained environment, we ran also a series of experiments where we decreased uniformly the number of flights in the area, without capacity constraint.

The results are presented in figure 6. The effect of sector

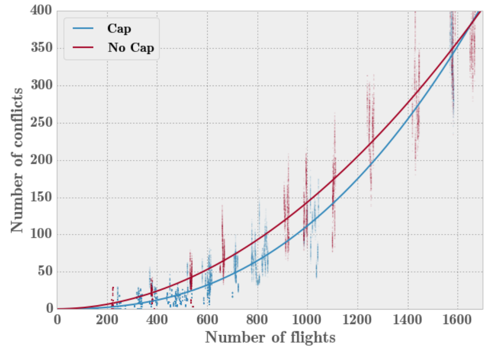

Fig. 6. Number of conflicts against the number of flights in the ACC, for capacity-constrained (Cap, blue) and unconstrained simulations (No Cap, red). In the constrained simulations, the sector capacities are uniformly increased to allow more and more flights to be flying. In the case of unconstrained simulations, the number of flights is directly controlled. Solid lines are the result of two-parameters power law fits.

capacity constraint is clearly visible. Except for very high numbers of flights, the number of conflicts is smaller when capacities are applied, as expected. Splitting the traffic among areas leads to decrease the overall complexity for a short-sighted controlling agent. It is striking though that the difference between the two cases becomes negligible at high traffic. In fact, a regression with a power law bN_f^a gives the following results:

- In the unconstrained case, $a = 2.0 \pm 9.10^{-5}$ and $b = 8.4\,10^{-5} \pm 3.10^{-11}$,
- In the constrained case, $a = 2.4 \pm 3.10^{-5}$ and $b = 5.0\,10^{-6} \pm 4.10^{-14}$

In the first case, the scaling is strikingly quadratic, in line with results of figures 2 and 3. In the second case the law is clearly superquadratic, which allows to have a slower increase for low traffic. However, a superquadratic law could be serious issue when the traffic becomes too high. This result can be easily interpreted.

Indeed, fractioning the airspace results in a decreased complexity within each sectors, but increased coordination is needed between sectors. In fact, one could see the current partition of the airspace as some kind of consensual optimal in regard of the trade-off between coordination and decreased complexity. More fractioning would require too much coordination, and more integration would increase the complexity too much. In our model, there is no explicit coordination, since there is a unique controller blind to the sectors (in these experiments at least). However, there is a similar effect. In fact, the workload of the controller decreases as long as the traffic is scarce enough to spread the traffic. But when the traffic is too high, sectors lose their independence anyway, because flights are present all over the airspace. As a consequence, the workload of the controller increases because instead of having a situation where all flights are taking more or less the same paths, it has to deal with all sorts of paths in the airspace, densely packed. The parallel with the real case is quite clear: to be able to split a sector in two parts, the resulting pieces of airspace need to gain some autonomy, i.e. the required coordination much be low enough.

Note that we obtain the same kind of results, with approximatively the same scaling in each case, when considering straight trajectories, i.e. in presence of free-routing.

VI. Conclusions and future work

We have presented an Air Traffic Simulator, developed within the ELSA project, that can be used to perform policy experiments relative to the SESAR scenario. The model describes either the strategic phase associated to the planning of the aircraft trajectories and the tactical modifications that might occur in the en-route phase.

In section V-A and section V-B we showed how in the free-route scenario we can expect that the controllers will have to perform a smaller number of operations although they will be more dispersed over a larger portion of the airspace. These results were obtained in the idealized case where capacity sectors do not play any role. Section V-A also showed the existence of a scaling law indicating that the number of operations performed by a controller quadratically depends from the number of aircraft present in the considered airspace. In section V-C we show that such quadratic scaling law is modified when the airspace is partitioned in air traffic sectors for which capacity constraints must be fulfilled. In this case the scaling law is still valid, although with a larger scaling exponent.

Other results obtained by using the ELSA Air Traffic Simulator were reported in Ref. [18]. Further research activities will investigate how the air traffic complexity is affected by the implementation of free-routing. From results obtained so far is in fact clear that the reduced number of conflicts comes to the price of an increased complexity due to the fact that they are spread on a larger area. The results of this research will be reported in future publications currently in preparation [19].

Moreover, the role of the time-horizon of the controller in an uncertain environment has been studied. The results show that the predictions of the controller can be destroyed by uncertainty, hence resulting in suboptimal decisions where a conflict resolution provokes more conflicts, or is totally useless. The “optimal” value of the time-horizon with respect to an expected level of noise on the trajectories can therefore be roughly evaluated with the model.

Acknowledgment

This work is co-financed by EUROCONTROL on behalf of the SESAR Joint Undertaking in the context of SESAR Work Package E - ELSA research project. The authors thank F. Lillo, R.N. Mantegna and S. Pozzi for fruitful discussion.

References

[1] SESAR, “Definition of the future atm target concept - d3,” 2007.
[2] “Commission regulation (EU) no 691/2010,” 2010.
[3] EUROCONTROL, “Final report on european commissions mandate to support the establishment of functional airspace blocks (fabs),” 2005.
[4] Complex World - SESAR WP-E Research Network for the theme “Mastering Complex Systems Safely” , *The ComplexWorld Position Paper* , Version: August 2012, (Restricted audience)
[5] EUROCONTROL, ”Free Route Developments in Europe”, February 2012.
[6] SESAR, ”SESAR Concept of Operations Step 1”, 2012.
[7] K. Tumer and A. Agogino, ”Distributed Agent-Based Air Traffic Flow Management”, In: Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems, May 2007.
[8] G. Granger, C. Allignol and N. Durand, ”The Influence of Uncertainties on Traffic Control using Speed Adjustments”, 9th USA/Europe ATM R&D Seminar, 2011.
[9] C. Allignol, N. Barnier, N. Durand and J-M. Alliot, ”A New Framework for Solving En-Route Conflicts”, 10th USA/Europe ATM R&D Seminar, 2013.
[10] D. Sislak, P. Volf, M. Pechoucek, C.T. Cannon , D. N. Nguyen and W. C. Regli, ”Multi-Agent Simulation of En-Route Human Air-Traffic Controller” Proceedings of the Twenty-Fourth Innovative Applications of Artificial Intelligence Conference, 2012
[11] Delaunay, B. 1934. Sur la sphère vide, *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk* **7** 793-800.
[12] Voronoi, Georgy (1908). “Nouvelles applications des paramètres continus la théorie des formes quadratiques”. *Journal fr die Reine und Angewandte Mathematik* 133 (133): 97–178. doi:10.1515/crll.1908.133.97
[13] C. Bongiorno, G. Gurtner, F. Lillo, L. Valori, M.Ducci, B. Monechi and S. Pozzi, An Agent Based Model of Air Traffic Management. In: Proceedings of The Third SESAR Innovation Days (2013) EUROCONTROL. 26th-28th November 2013, Stockholm, Sweden. edited by D. Schaefer
[14] ELSA project, *E.02.18-ELSA D2.3 Calibrated agent-based model final draft*, Version: March 2014, (Restricted audience)
[15] G. Gurtner, L. Valori, F. Lillo, Competitive allocation of resources on a network: an agent-based model of air companies competing for the best routes JSTAT **2015**(5), P05028 (2015)
[16] G. Van Rossum; F. L. Drake Jr. *Python reference manual*. (Centrum voor Wiskunde en Informatica, Netherlands, 1995).
[17] B. W. Kernighan; D. M. Ritchie. *The C Programming Language* . (Prentice Hall, England, 1988).
[18] ELSA project, *E.02.18-ELSA D2.4 SESAR agent-based model*, Version: June 2015, (Restricted audience)
[19] S. Miccich, C. Bongiorno, G. Gurtner, M. Ducci, *An Empirically grounded Agent Based Model for the SESAR scenario*, in preparation.