

DESCRIPTION OF THE AGENT-BASED MODEL OF ELSA

Empirically grounded agent based models for the future ATM scenario

The agent-based model developed by the ELSA project aims at simulating the European air traffic. It is a versatile model which can be used as a library to develop other types of models easily by making use of different parts independently. Some preliminary results obtained with this model can be found in [1].

1 Overview

This model is based on interacting agents (mainly flights and controllers) and is composed of several fairly independent modules:

- A network generator, including navpoints and sectors.
- A trajectory generator, used to test different traffic situations.
- A tool to “rectify” trajectories in case of simulation of the SESAR scenario,
- A conflict resolution engine, simulating a tunable, imperfect controller.
- A post-processing module, including standard metrics computation and a simple graphic interface to see isolated run.

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 define in which area the controller can actually interact with the flight.

The core engine of the model is the resolution of conflicts, given an airspace and some flights travelling through the sector. 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 he 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 SESAR 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.

2 Features

The resolution engine itself is written in C, with an interface with Python. The code is organized in a modular way, making it possible to use only the resolution engine and feeding it with some user defined airspace, or use only the trajectory generator, etc. Moreover, some default post-processing manipulations are included for simple metrics about the networks, on the trajectories and on the actions of the controller.

2.1 Network generator

The network generator module is very versatile and allows:

- To draw the spatial distribution of navigation points or use external data,
- To compute the navigation points network edges with a triangulation¹ or use external data,
- To draw sectors at random, using a Voronoi tessellation for the boundaries or use external data.
- To compute time of travels between edges of navigation points or use external data.

Hence 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.

This module is written in Python.

2.2 Trajectories generator

The trajectories generator² can be used to generate synthetic traffic on a given network of navigation points + 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:

- A total number of flights,

¹We use the Delaunay triangulation for its properties [2].

²It is part of a bigger model called “Strategic Model” which was used by ELSA to study the strategic phase of the Air Traffic Management. The full strategic model will also be available in the same repository and is written in Python [4].

- a distribution of flights per pair of entry/exit points,
- some capacities for the sectors.

This module is written in Python.

2.3 Trajectories rectification

With this module, one can study the trajectory-based operations in the SESAR scenario. In the SESAR Concept of Operation [5] is in fact foreseen the implementation of Business Trajectories where requested trajectories from airlines will be progressively made straighter and straighter across sectors and FIRs. With these trajectories, the flights are not bound to travel on the navpoint network anymore, but can rather use straighter trajectories.

This rectification happens for trajectories at random, moving one navpoint chosen at random on the trajectory to make it straighter. Different probabilities can be associated to different sets of navpoints, hence allowing heterogeneous transition to business trajectories, for instance for different sectors.

The navpoints are rectified until a certain value of the “efficiency” is reached. The efficiency is the average ratio over all flights of the length of the direct trajectory (from entry point to exit point) with the current trajectory. This efficiency is directly related to the Key Performance Indicator defined by the Performance Review Commission [6] as “flight efficiency”.

This module is written in Python.

2.4 Conflict resolution engine

This engine is used to deconflict a set of flights on an airspace. A “super-controller” is trying to avoid conflicts by changing the trajectories of flights using the three dimensions. Some features included in the engine are:

- The controller is actually deconflicting all the flights: no conflict ever occurs,
- However, the controller is not perfect because she has a limited lookahead time. Hence, she takes actions which could trigger other conflicts in the future.
- Some noise can be applied on the speed of the aircraft, possibly destroying the forecast of the controller.
- The user can trigger some “shocked areas”, volume of the airspace forbidden to any flight. This is simulating closed airspace for military reasons or local weather events.

This generator also includes a (tactically) pre-deconflicting module (right now, a brute force algorithm shifting flights in time). This can be used to study the case where only the noise on velocity (i.e., predictability) impacts the conflict resolutions. Note that in the current scenario, the planned trajectories are NOT conflict-free. Thus, the

pre-deconflicting module aims at simulating the improved coordination, information sharing and trajectory prediction that is foreseen in the SESAR scenario [5].

This module is written in C with an interface with Python.

2.5 Post-Processing tools

Some post-processing tools are also included in the library, including:

- Different graphs showing statistics on networks and trajectories,
- community detection tools using (scalable) modularity, infomap and OSLOM,
- graphic interface to replay single runs.

References

- [1] G. Gurtner, F. Lillo, L. Valori, M. Ducci, B. Monechi, S. Pozzi, C. Bongiorno, S. Miccichè, R. N. Mantegna, “An Agent Based Model of the Air Traffic Management”, *Proceedings of the SESAR Innovation Days*, 2014.
- [2] Delaunay, B. 1934. Sur la sphère vide, *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk* **7** 793-800.
- [3] Voronoi, Georgy (1908). “Nouvelles applications des paramètres continus à la théorie des formes quadratiques”. *Journal für die Reine und Angewandte Mathematik* 133 (133): 97–178. doi:10.1515/crll.1908.133.97
- [4] Gurtner, G, Valori, L., and Lillo F. 2014, Competitive allocation of resources on a network: an agent-based model of air companies competing for the best routes, *submitted to JSTAT*, 2014.
- [5] SESAR, “SESAR Concept of Operations Step 1” 2012.
- [6] Performance Review Commission, Performance Review Report 2013.