Does ATM Need Centralized Coordination? Autonomous Conflict Resolution Analysis in a Constrained Speed Environment

Nicolas Durand, Nicolas Barnier

Does ATM Need Centralized Coordination? Autonomous Conflict Resolution Analysis in a Constrained Speed Environment

Nicolas Durand and Nicolas Barnier
ENAC/MAIAA
7, av. Édouard Belin
31 055 Toulouse, France
firstname.surname@enac.fr

***Abstract*—The concept of Free-Flight, introduced in the 90s, opened a debate on the efficiency of letting aircraft deal with conflicts without any centralized control. Many models have been proposed for autonomous aircraft solvers but their efficiency is not well-known. In this paper, we experiment powerful algorithm derived from robotics which is able to deal with thousands of robots in very small spaces, and show how its performance plummets when speeds are constrained. We also compare this autonomous algorithm with a centralized approach using evolutionary computation on a complex example to point out their relative performance in a speed constrained environment. This comparison provides scientific arguments for the necessity of centralized air traffic control.**

Keywords: conflict resolution, free-flight, geometrical algorithms, evolutionary computation

INTRODUCTION

When the concept of Free-flight emerged in the 90s, a large debate started on the possibility of eliminating centralized control systems by developing autonomous algorithms for solving conflicts. The first effective approach used sliding forces to coordinate maneuvers between aircraft [1]. Potential or vortex fields [2] as well as a model based on an analogy with electrical particle repulsion [3] were also used.

However, these approaches were never compared to a centralized method and the behaviors of the algorithms were not tested in very dense environments. They also needed to be tuned carefully to be effective and finding general rules to do so did not seem possible.

In 2001, we proposed a token allocation strategy combined with an A^* algorithm to solve conflicts with realistic maneuvers [4], [5]. Even if some maneuvers could be simultaneously decided, a complete ranking of aircraft was necessary and finding an optimal ranking has been shown to be problem dependent [6]. We also tried neural networks on the two-aircraft problem [7] but they could not be generalized to handle more aircraft.

Geometrical approaches have also been widely studied [8]–[11]. The powerful technique developed by Van den Berg et al. [11] can handle thousands of agents in a small space. It was applied to aircraft by Snape et al. [10], but the hypotheses of the algorithm require simultaneous vertical and horizontal speed changes, which are not realistic.

In the 90s, we showed that conflict resolution was highly combinatorial and could not be globally solved with local optimization methods [12]. We proposed a genetic algorithm to solve multiple aircraft conflicts using a centralized approach with simple maneuvers (similar to those used by air traffic controllers) and showed that we were able to solve every conflict on real traffic data [13].

Centralized methods can be divided into two main categories. Some methods [14]–[16] use greedy sequential algorithms to optimize trajectories one by one after ranking the aircraft. However, finding an appropriate ordering is challenging [6]. The others, mentioned in the next paragraph, try to find the global optimum without the need to prioritize aircraft.

Using evolutionary computation, our team was the first to address conflict resolution globally [12]. Others later introduced a powerful Semidefinite Programming approach [17], [18], which is also able to handle multiple aircraft in a single scenario. However the solution is only locally optimal and the model requires constant speeds and perfect trajectory prediction. In the early 2000s, a mathematical model using Mixed Integer Linear Programming, which could be solved by CPLEX and ensured the global optimality of the solution, was proposed [19], [20]. The model was extended in 3D in 2008, but required constant speed during climbing phases [21]. Uncertainties on the trajectory heading were added to the horizontal model in 2009, but all maneuvers still needed to be executed at the same time (at every optimization step) [22]. The method is powerful but cannot be used for developing a realistic advisory tool for controllers. Finally in 2013, our team used a Constraint Programming approach to globally solve complex conflicts [23]. For each aircraft, a number of alternative trajectories and a matrix of pairwise conflicts were precalculated, taking various uncertainties into account, before the optimization process was performed.

In this study, we use the autonomous approach proposed by Van den Berg et al. in [11]. We relax some realistic hypotheses on trajectories and revert to a simple horizontal model in order to assess the efficiency of an autonomous conflict resolution solver for aircraft with various levels of maneuverability in the horizontal plane. In section I of this article, we detail the geometrical approach developed by Van den Berg et al. [11]. Then we test it in a dense environment in section II and

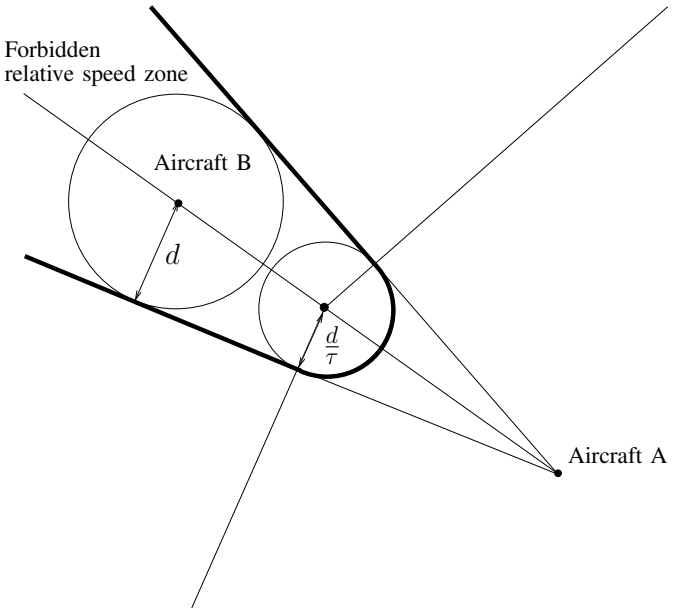

Fig. 1. Conflicting aircraft model: a conflict will occur within time τ if and only if the relative speed $\vec{v_r}$ lies in the forbidden zone.

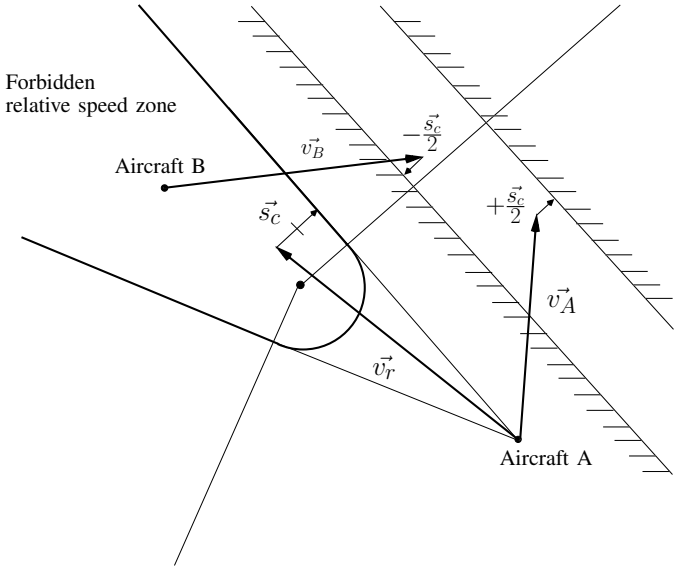

Fig. 2. Conflict resolution model: the necessary speed change to move $\vec{v_r}$ out of the forbidden zone is divided in two and shared by both aircraft.

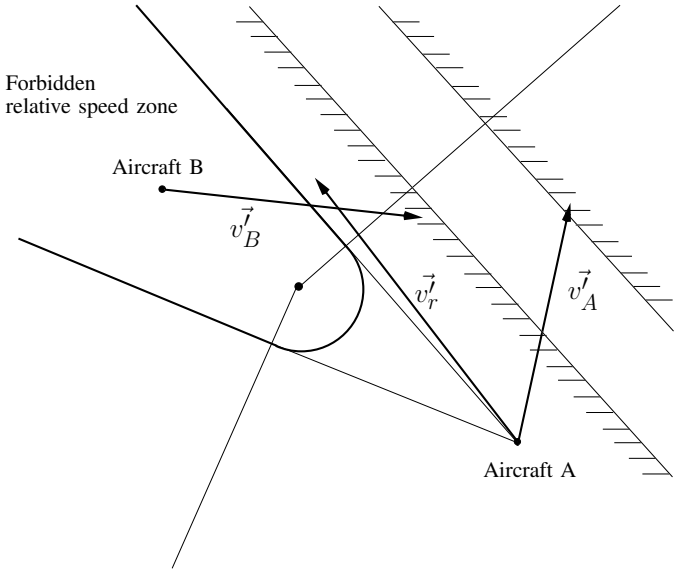

Fig. 3. Solved conflict: updated speeds $\vec{v'_A}$ and $\vec{v'_B}$ lie outside of the conflict zone.

constrain the aircraft speeds and turning angles to show how the performance of the algorithm changes. In section III, we compare the autonomous algorithm to a centralized approach based on an Evolutionary Algorithm and use an example to show the advantage of coordinating maneuvers when the speed is constrained. The last section discusses the consequences of speed constraints on the autonomous solver and highlights directions for future work.

I. Geometrical Model

This section describes the algorithm developed in [11]. Let d be the standard separation between aircraft and τ be the lookahead time. In figure 1, let us consider two aircraft A and B. We can represent the position of aircraft B in the referential of aircraft A. If we draw a circle of radius d around aircraft B, the two lines issued from position A, tangent to the circle of radius d form a cone. If the relative speed $\vec{v_r} = \vec{v_A} - \vec{v_B}$ lies in this cone, a conflict will occur in the future. If we draw a circle of size $\frac{d}{\tau}$ tangent to the two previous lines, we obtain a new zone bounded by the bold line in figure 1. It is then straightforward to understand that a conflict will occur within time τ if and only if $\vec{v_r}$ lies in this zone.

When a potential conflict occurs (i.e. $\vec{v_r}$ is in the forbidden zone), $\vec{v_r}$ is projected on the nearest bold line to escape the forbidden zone (see figure 2) and solve the conflict pairwise between aircraft A and B. The necessary speed change $\vec{s_c}$ to move $\vec{v_r}$ out of the forbidden zone is then divided in two and shared by both aircraft.

We can then define two half-planes limited by lines perpendicular to $+\frac{\vec{s_c}}{2}$ and $-\frac{\vec{s_c}}{2}$: if the new values of $\vec{v_A}$ and $\vec{v_B}$ are chosen in their respective half-plane, then the new relative speed $\vec{v_r}$ is out of the forbidden speed zone. An example of the modified speeds is shown in figure 3.

In order to take into account more than two aircraft, the model can be extended. For each pair of aircraft, a half-plane is created. In figure 4, the intersection of the half-planes defined by aircraft B, C and D using the previous principle represents the acceptable speed zone for aircraft A.

Because the possible maneuvers of an aircraft are heavily constrained by aerodynamic laws and operations rules, the speed vector change at each time step is limited in our model. For the sake of simplicity, we can choose to change the aircraft speed in a box of length $2\,s_{\mathrm{lon}}$ and of width $2\,s_{\mathrm{lat}}$ (see figure 5). The intersection of this box and the previous area defines a permitted speed zone for aircraft A. If we project the optimal speed (heading to the destination) on this zone we can define the new speed for aircraft A.

If the permitted speed zone is empty, we uniformly increase the acceptable speed zone until the intersection with the speed change box is not empty. This new speed, although it no longer

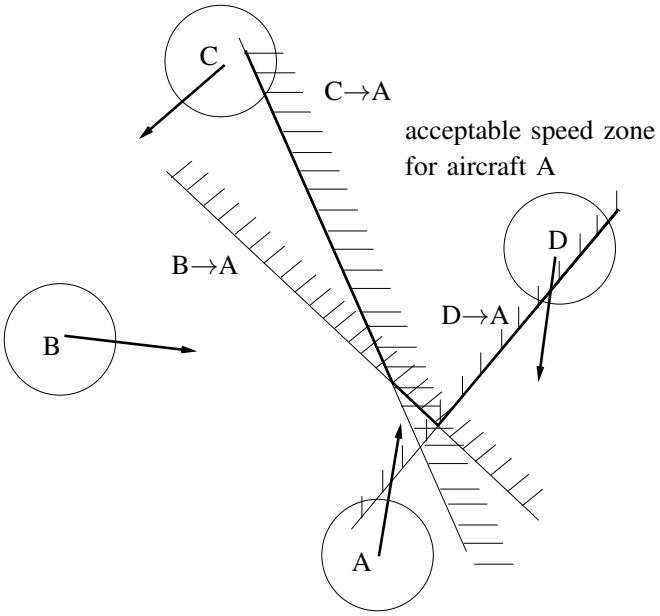

Fig. 4. Acceptable speed zone for aircraft A.

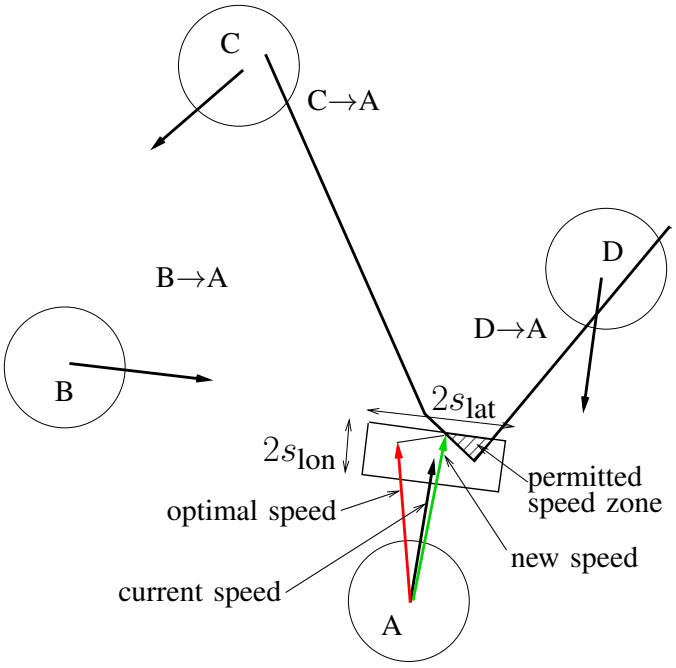

Fig. 5. New speed for aircraft A in the permitted speed zone (dashed).

guarantees zero conflict for τ seconds, represents a satisfactory solution, i.e. with the least conflicts.

The same algorithm can be extended to avoid obstacles such as sector edges and restricted (e.g. military) zones. In this particular case, the speed change is not divided but totally allocated to the aircraft.

II. Experimental results with various speed ranges

The previous algorithm has proved to be very efficient in robotics [11]. It is also successfully used to mimic crowd

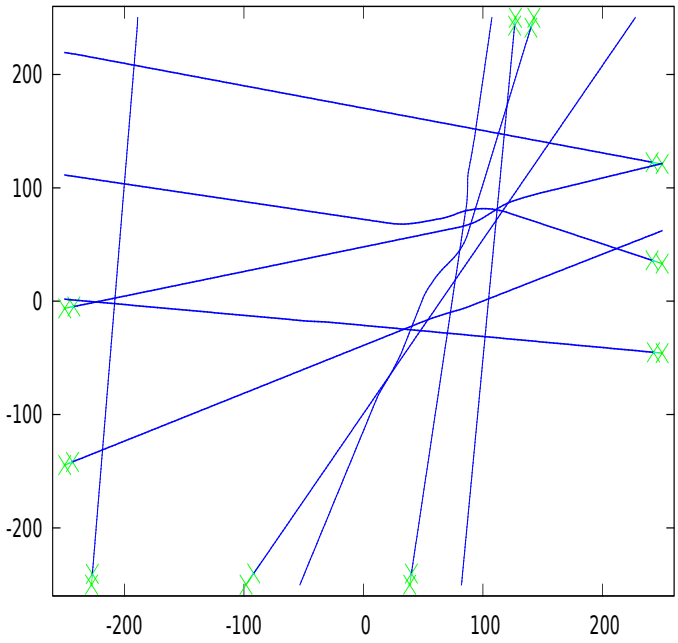

Fig. 6. A scenario example with 10 aircraft.

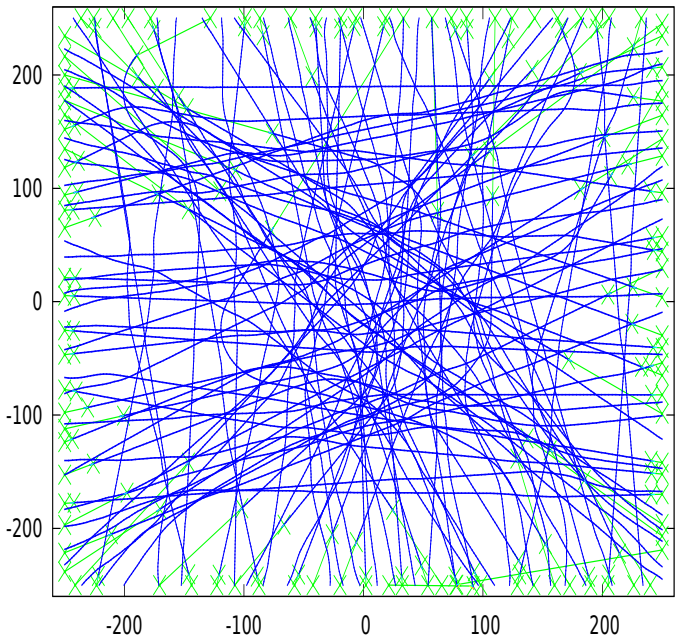

Fig. 7. A scenario example with 100 aircraft.

movements or building evacuations. It is easy to parallelize and can thus handle thousands of objects at the same time.

However, we need to assess its behavior in a speed constrained environment to check if it could be applied to air traffic conflict resolution. Therefore, we test it in a 500×500 nautical mile sector with varying numbers of aircraft n and different parameters for s_{lat} and s_{lon}. We only focus on horizontal maneuvers in a single flight level. In the following, low density is defined as 10, 20, 30 aircraft, medium as 50 aircraft, and high as 100, 120 or 130 aircraft. Figure 6 and 7 show the resulting radar tracks for two instances with 10 and 100 aircraft respectively.

A. *Influence of the speed constraint* s_{lon}

The aircraft speeds are set to 1 (speeds are normalized), which means that we first set all aircraft to fly at the same speed. This is not totally realistic, but similar results are obtained with nominal speeds varying by $\pm 20\%$. The time step is set to 3 seconds. For a standard turning rate of 3 degrees per second we round the value of s_{lat} at 0.05. We fix $\tau = 360\,s$ (i.e. 6 min) because it is a good compromise: if the lookahead time is too long, the algorithm is less efficient for high densities because too many conflicts are simultaneously taken into account and the permitted zones often are empty; whereas if τ is too short, some conflicts appear too late to be resolved.

We first run 100 executions for different values of n and s_{lon} and for $s_{\text{lat}} = 0.05$. For each execution, the origins and destinations of the aircraft are randomly chosen on the edges of the square. We make sure that aircraft are separated by at least 2 separation standards (10 NM) at the origin and the destination. A quarter of the traffic goes from the east side to the west side, another quarter from west to east, a quarter from north to south and the last one from south to north.

For each scenario (n, s_{lon}), we first count the number of runs which fail because aircraft are "pushed" out of the 500×500 nautical mile sector before reaching their destination. We then count the number of runs which do not solve every conflict. For the remaining runs, we measure the trajectory lengthening (the ratio of the modified trajectory and the original trajectory).

Table I gives the experimental results for densities varying from 10 to 50 aircraft and various s_{lon} values representing the speed range. E.g. $s_{\text{lon}} = 0.01$ means that the updated aircraft speed can be chosen within $\pm 1\%$ of the current speed, whereas with $s_{\text{lon}} = 1.$, the new speed can be within $\pm 100\%$ of the former one. The latter is totally unrealistic for aircraft but is tested to compare the efficiency of the algorithm for different speed ranges. Table II gives results for higher densities varying from 100 to 130 aircraft and various s_{lon} values.

When s_{lon} becomes very small, aircraft can only avoid each other by changing headings. For converging aircraft with close headings, aircraft tend to choose parallel routes and postpone conflicts instead of solving them. This is an issue because aircraft slowly get away from their original tracks and reach the edge of the sector without solving conflicts. When such a border violation occurs, the instance is considered a failure.

nb acft	s_{lon}	Tot Fail	Border Viols	Rem Conf	mean delay	max delay
10	0.0001	46	44	2	1.006094	1.078254
10	0.001	46	43	3	1.005470	1.132507
10	0.01	29	28	1	1.004209	1.025716
10	0.02	21	19	2	1.003344	1.034413
10	0.03	12	12	0	1.002389	1.018578
10	0.04	7	7	0	1.002415	1.025875
10	0.05	2	2	0	1.000944	1.008980
10	0.08	1	1	0	1.000325	1.003012
10	0.1	0	0	0	1.000172	1.001859
10	1.	0	0	0	1.000152	1.001391
20	0.0001	97	83	14	1.006339	1.013132
20	0.001	93	87	6	1.010896	1.023264
20	0.01	85	80	5	1.006144	1.026616
20	0.02	74	71	3	1.010969	1.033025
20	0.03	63	62	1	1.007849	1.031465
20	0.04	29	25	4	1.004676	1.020729
20	0.05	15	13	2	1.004316	1.030199
20	0.08	2	0	2	1.001048	1.005255
20	0.1	0	0	0	1.000714	1.008693
20	1.	0	0	0	1.000398	1.001875
30	0.01	100	86	14		
30	0.02	94	81	13	1.009414	1.020900
30	0.03	89	83	6	1.013810	1.035538
30	0.04	67	60	7	1.006878	1.017599
30	0.05	41	39	2	1.007275	1.023955
30	0.08	7	3	4	1.002129	1.007207
30	0.1	2	1	2	1.001521	1.016695
30	0.2	0	0	0	1.000656	1.002959
30	0.5	1	0	1	1.000643	1.002853
30	1.	0	0	0	1.000702	1.002278
50	0.01	100	50	50		
50	0.02	100	68	32		
50	0.03	100	72	28		
50	0.04	97	75	22	1.018516	1.027195
50	0.05	90	74	16	1.013760	1.033422
50	0.08	31	23	8	1.006376	1.021587
50	0.1	9	6	3	1.004415	1.022348
50	0.15	3	0	3	1.002292	1.015073
50	0.2	1	0	1	1.001874	1.007279
50	0.5	0	0	0	1.001677	1.004159
50	1.	0	0	0	1.001650	1.003775

TABLE I
TOTAL NUMBER OF FAILURES, DUE TO BORDER VIOLATIONS OR REMAINING CONFLICTS, MEAN AND MAX DELAYS FOR 10, 20, 30 OR 50 AIRCRAFT AND VARIOUS SPEED RANGE s_{lon} VALUES.

Figure 8 shows the total number of failures for different densities of traffic as a function of the speed range $s_{\text{lon}} \in [0., 1.]$. For the low and medium densities (10, 20, 30 or 50 aircraft) the algorithm is very efficient when the speed can be modified by at least 10% but suddenly becomes totally ineffective when the speed is constrained in a smaller interval. This is also true in the high density ranges: for 100 aircraft the solver becomes efficient for speed changes greater than 15%, which is still low. Even for the densest cases (120 and 130 aircraft), a speed change of 25% is enough to solve the majority of cases.

These figures give a better understanding of the efficiency of such algorithms in robotics where agents can change speeds very easily. Autonomous approaches also works very well to model car or pedestrian traffic for the same reason. These

nb acft	s_{lon}	Tot Fail	Border Viols	Rem Conf	mean delay	max delay
100	0.05	100	8	92		
100	0.08	100	32	68		
100	0.09	98	34	64	1.024838	1.032395
100	0.1	95	36	59	1.019754	1.026248
100	0.12	78	37	41	1.014380	1.027306
100	0.15	47	19	28	1.011587	1.024159
100	0.17	36	17	19	1.011604	1.027599
100	0.2	25	13	11	1.010599	1.020105
100	0.25	15	5	10	1.008623	1.019754
100	0.3	7	2	5	1.008284	1.020691
100	0.35	3	0	3	1.008063	1.019256
100	0.4	2	1	1	1.007393	1.015109
100	0.45	2	0	2	1.007272	1.011421
100	0.5	2	0	2	1.007192	1.017652
100	0.7	2	0	2	1.007000	1.012067
100	1.	2	1	1	1.006742	1.018752
120	0.05	100	0	100		
120	0.08	100	5	95		
120	0.1	100	8	92		
120	0.12	98	16	82	1.023625	1.031094
120	0.15	90	24	66	1.018168	1.027560
120	0.2	65	12	53	1.015557	1.024916
120	0.25	36	7	29	1.013526	1.028587
120	0.3	21	2	19	1.012241	1.021002
120	0.35	11	2	9	1.011288	1.021665
120	0.4	5	2	3	1.010339	1.022460
120	0.5	3	0	3	1.010307	1.017107
120	1.	5	1	4	1.009770	1.014701
130	0.05	100	0	100		
130	0.08	100	0	100		
130	0.1	100	6	94		
130	0.12	99	5	94	1.025159	1.025159
130	0.15	94	14	80	1.021134	1.029294
130	0.2	72	15	57	1.016474	1.034786
130	0.25	49	3	46	1.015537	1.024774
130	0.3	25	2	23	1.013926	1.025481
130	0.35	18	3	15	1.012957	1.020840
130	0.4	15	2	13	1.013010	1.024330
130	0.5	11	0	11	1.012406	1.023198
130	0.7	9	0	9	1.011538	1.019124
130	1.	6	1	5	1.011229	1.019580

TABLE II
TOTAL NUMBER OF FAILURES, DUE TO BORDER VIOLATIONS OR REMAINING CONFLICTS, MEAN AND MAX DELAYS FOR 100, 120 OR 130 AIRCRAFT AND VARIOUS SPEED RANGE s_{lon} VALUES.

results show as well how such algorithms lose their efficiency when the speed is constrained. In particular, the European ERASMUS project [24], [25], which aims at automatically solving conflict by slight speed changes, set that $[-3\%, +6\%]$ was a reasonable speed change range. With such a short range, the autonomous approach described here can only handle very low densities, even if aircraft are allowed to change heading (whereas it was not the case for the ERASMUS project).

Figure 9 zooms in on the speed change ranges that are acceptable for aircraft, which is about $\pm 5\%$. It is noticeable that only the lowest density allows a solution for every conflict in this range. We can also remark that with no speed change, only half of the scenarios are solved with the lowest density experimented (10 aircraft).

When we look at the reasons for failure, in very low

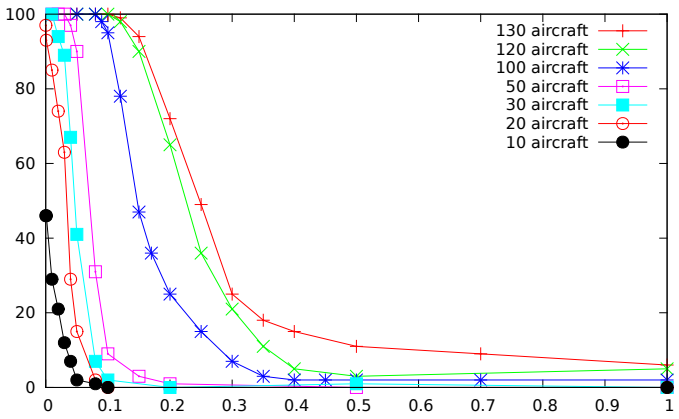

Fig. 8. Total number of failures for different densities of aircraft as a function of the speed range $s_{\text{lon}} \in [0., 1.]$.

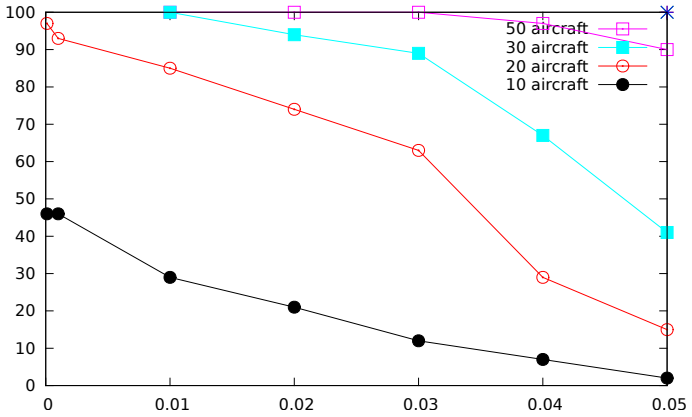

Fig. 9. Total number of failures for 10, 20, 30 or 50 aircraft as a function of the speed range $s_{\text{lon}} \in [0., 0.05]$.

densities without speed change ($s_{\text{lon}} \leq 0.1$), most of them are border violations, which means that the autonomous approach fails to find a solution that could keep every aircraft inside the designed sector. This is even the case with only 10 aircraft for half of the runs when no speed change is allowed.

Figure 10 and 11 show the number of failures due to border violations and those due to remaining conflicts. For high densities ($\geq$ 100 aircraft), the failures are mainly due to remaining conflicts. For low densities ($\leq$ 30 aircraft) the failures are mainly due to border violations, because there is enough space to keep aircraft separated, but they tend to "push" each other out of the sector.

Figure 12 shows the trajectory lengthening in the different cases. It increases with the density, which is expected, and decreases when the speed constraint loosens because aircraft can accelerate and compensate for the delays caused by conflicts.

B. *Influence of the turning angle constraint* s_{lat}

When $s_{\text{lon}} = 0$, s_{lat} can be seen as the tangent of the turning angle. In this section, we set $s_{\text{lon}} = 0.01$ (the speed change cannot exceed 1%) and assess the behavior of the autonomous algorithm for different densities and values of s_{lat}. Table III

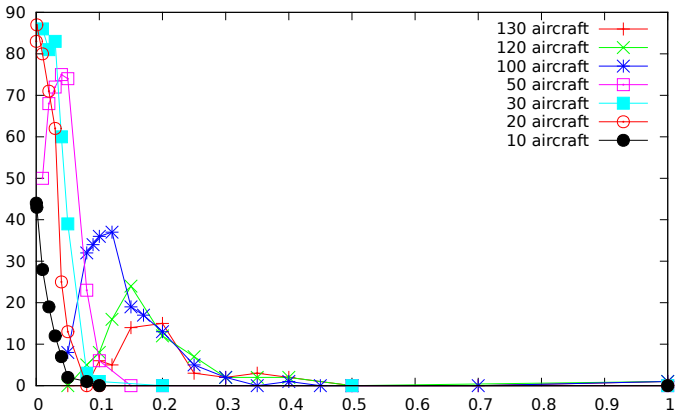

Fig. 10. Failures due to border violations (as a function of $s_{\text{lon}} \in [0., 1.]$).

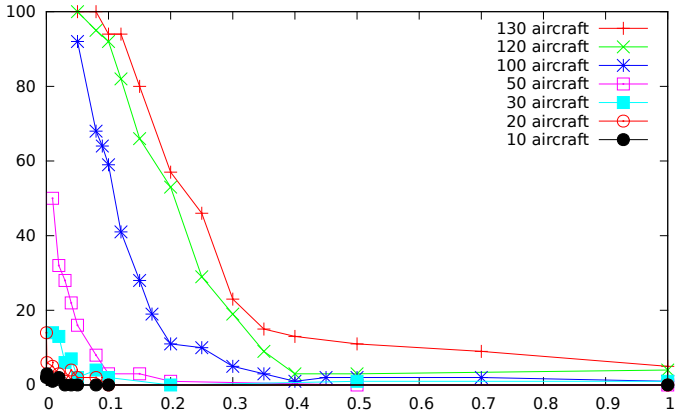

Fig. 11. Failures due to remaining conflicts (as a function of $s_{\text{lon}} \in [0., 1.]$).

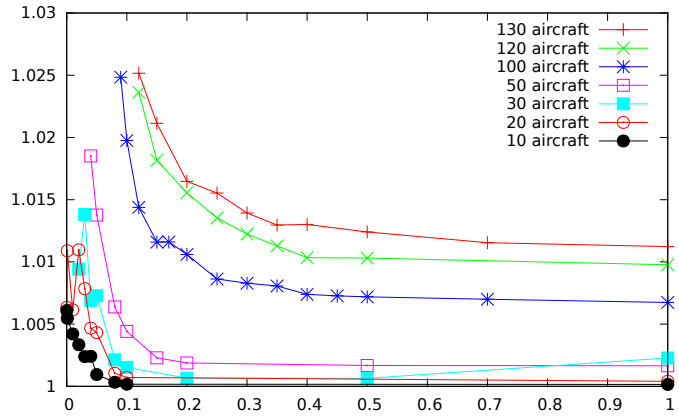

Fig. 12. Trajectory lengthening (as a function of $s_{\text{lon}} \in [0., 1.]$).

nb acft	s_{lat}	Tot Fail	Border Viols	Rem Conf	mean delay	max delay
10	0.01	49	11	38	1.000273	1.002032
10	0.05	33	32	1	1.005172	1.060118
10	0.10	19	19	0	1.002365	1.023111
10	0.20	18	18	0	1.002842	1.022097
10	0.30	12	12	0	1.004603	1.070145
10	0.40	7	7	0	1.004182	1.070736
10	0.50	12	10	2	1.007833	1.070753
10	0.60	9	9	0	1.006707	1.125945
10	0.70	7	7	0	1.008278	1.247765
10	0.80	5	5	0	1.006431	1.139933
10	0.90	8	8	0	1.009116	1.138795
10	1.00	2	2	0	1.009575	1.153068
20	0.01	94	8	86	1.000714	1.001241
20	0.05	80	75	5	1.011828	1.030637
20	0.10	70	70	0	1.008321	1.044828
20	0.20	49	49	0	1.009260	1.043033
20	0.30	38	37	1	1.009096	1.067048
20	0.40	47	47	0	1.013962	1.051846
20	0.50	32	31	1	1.017457	1.085313
20	0.06	28	28	0	1.017957	1.164125
20	0.70	23	22	1	1.018517	1.101829
20	0.80	25	24	1	1.015956	1.105185
20	0.90	18	15	3	1.018290	1.102465
20	1.00	24	23	1	1.023560	1.154436
30	0.01	100	1	99		
30	0.05	100	89	11		
30	0.10	95	92	3	1.012198	1.022582
30	0.20	87	87	0	1.019120	1.037669
30	0.30	77	77	0	1.017174	1.035909
30	0.40	59	59	0	1.026756	1.110479
30	0.50	60	59	1	1.030889	1.104633
30	0.60	47	43	4	1.035111	1.099216
30	0.70	49	47	2	1.035078	1.107805
30	0.80	45	41	4	1.036041	1.121747
30	0.90	43	38	5	1.029661	1.091537
30	1.00	38	35	3	1.031138	1.102045
50	0.01	100	0	100		
50	0.10	100	84	16		
50	0.20	100	95	5		
50	0.30	99	94	5	1.068680	1.068680
50	0.40	94	89	5	1.060271	1.086596
50	0.50	88	84	4	1.055564	1.102381
50	0.60	81	78	3	1.062259	1.110347
50	0.70	86	78	8	1.059633	1.083060
50	0.80	82	75	7	1.061017	1.117620
50	0.90	72	61	11	1.058173	1.109486
50	1.00	75	58	17	1.063235	1.168941
100	0.01	100	0	100		
100	1.00	100	0	100		

TABLE III
TOTAL NUMBER OF FAILURES, DUE TO BORDER VIOLATIONS OR REMAINING CONFLICTS, MEAN AND MAX DELAYS FOR 10, 20, 30 50 OR 100 AIRCRAFT AND VARIOUS s_{lat} VALUES.

shows the experimental results for densities varying from 10 to 100 aircraft and various s_{lat} values.

Figure 13 shows the number of total failures for different densities as a function of s_{lat}. For high densities, no solution is found, whatever the turning angle. For medium density (50 aircraft) the number of failures slowly decreases when s_{lat} increases but it stays high (≥ 70). For low densities, the number of failures also decreases but never reaches zero.

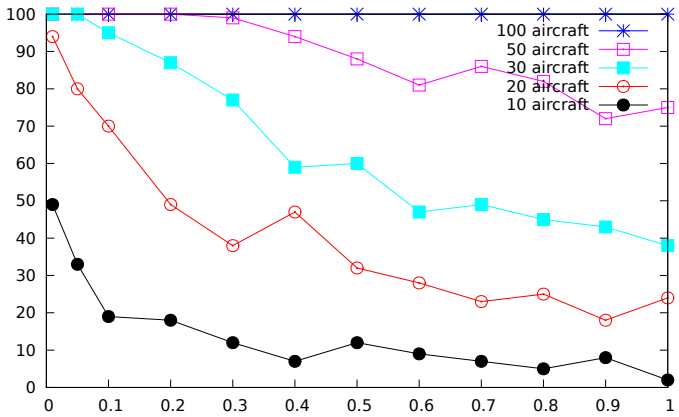

Fig. 13. Total number of failures for different densities of aircraft and various angles ($s_{\text{lat}} \in [0., 1.]$).

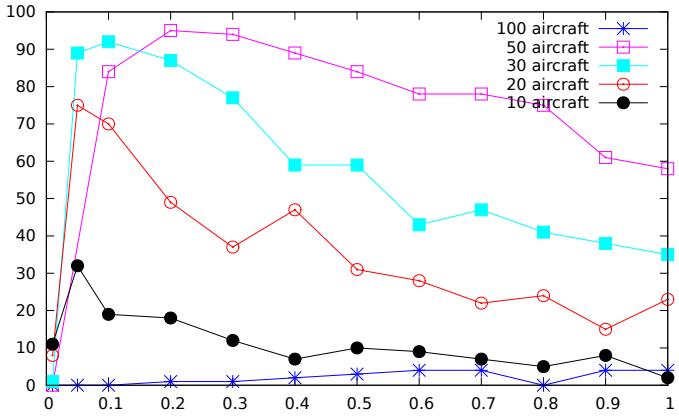

Fig. 14. Failures due to border violations (function of $s_{\text{lat}} \in [0., 1.]$).

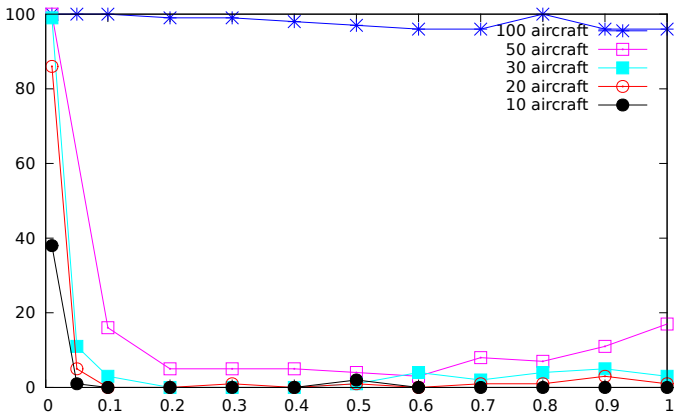

Fig. 15. Failures due to remaining conflicts (function of $s_{\text{lat}} \in [0., 1.]$).

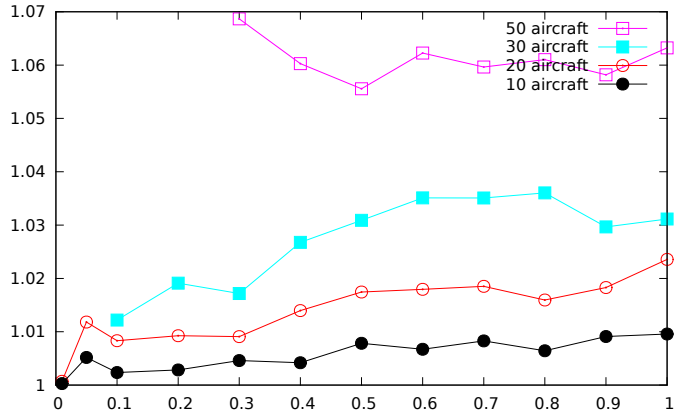

Fig. 16. Trajectory lengthening (function of $s_{\text{lat}} \in [0., 1.]$).

Figure 14 (resp. 15) shows the number of failures due to border violations (resp. remaining conflicts). For high densities, remaining conflicts cause almost every failure. For medium and low densities, when $s_{\text{lat}} \leq 0.05$, conflicts are mostly due to remaining conflicts, whereas when $s_{\text{lat}} > 0.05$ border violations cause most of the failures. These results show that increasing the lateral maneuverability s_{lat} helps solve more conflicts for low and medium densities, but it tends to "push" aircraft outside the borders of the sector. It does not help the autonomous algorithm to solve high density scenarios.

Finally, figure 16 shows the mean trajectory lengthening as a function of s_{lat}. As expected, the lengthening increases both with the traffic density and s_{lat}.

III. Comparison with a centralized approach on an example

In this section, we use an example to compare the autonomous algorithm to a centralized approach based on an Evolutionary Algorithm, in order to show the necessity to coordinate maneuvers when the speed is constrained. The algorithm used is the same as described in [13]. We changed the modeling in order to allow the same continuous maneuvers as for the autonomous approach. At each time step, an evolutionary algorithm is used to optimize the speeds that guarantee no conflict for τ seconds, and check the effectiveness of the resolution (verifying that aircraft that need to pass each other really do).

The example shows 15 aircraft converging on the center of a circle (with a 5 degree angle of convergence between aircraft). Figure 17 shows the solution found with the autonomous algorithm and no speed change allowed. Figure 18 shows the solution found with the autonomous algorithm and 20% speed change allowed. Figure 19 shows the solution found with the centralized approach and no speed change allowed. Here we can observe the benefit of using centralized information to coordinate maneuvers and help aircraft pass each other without slowing or accelerating them.

Conclusion and Future Work

In this article, we presented an autonomous conflict resolution algorithm derived from robotics and able to handle thousands of agents when speeds and turning angles of the robots are not constrained. Our experiments showed how the speed constraint severely degrades the algorithm's performance, even when the density of traffic is low. Even in low densities, some scenarios failed to find solutions when the speed was constrained: aircraft converging with a small angle tend to "push" each other out of the sector. In contrast, the turning angle constraint has less of an impact on the resolution.

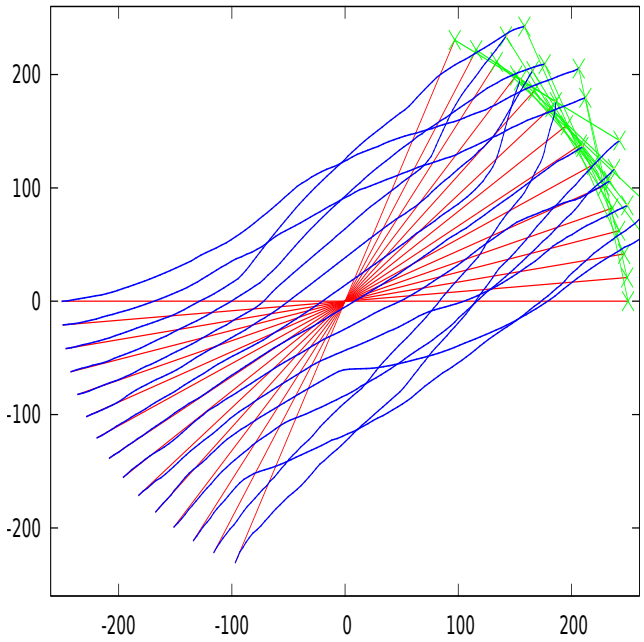

Fig. 17. 15 converging aircraft: reactive resolution with $s_{\mathrm{lat}} = 0.2$ and $s_{\mathrm{lon}} = 0.0001$ (no speed change allowed).

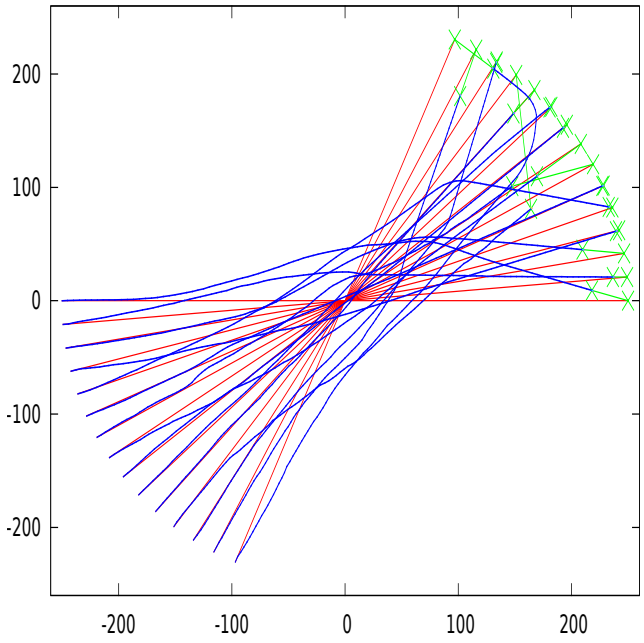

Fig. 18. 15 converging aircraft: reactive resolution with $s_{\mathrm{lat}} = 0.2$ and $s_{\mathrm{lon}} = 0.2$ (20% speed change allowed).

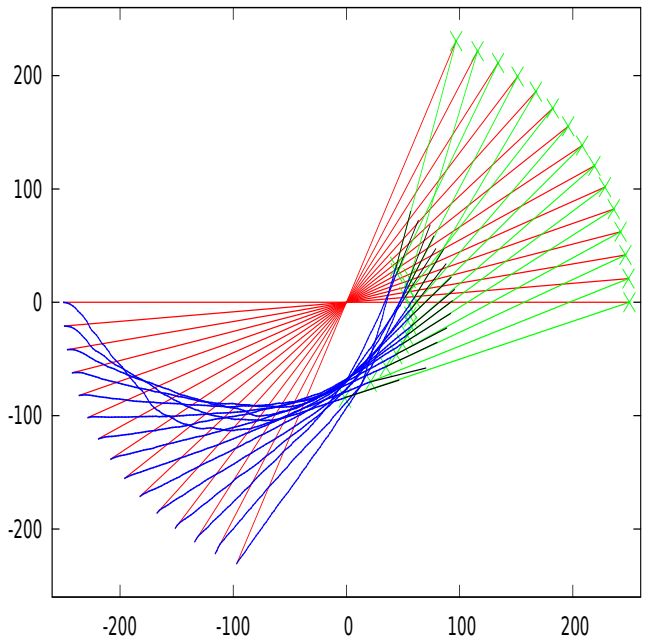

Fig. 19. 15 converging aircraft: centralized resolution with $s_{\mathrm{lat}} = 0.2$ and $s_{\mathrm{lon}} = 0.0001$ (no speed change allowed).

We designed a particularly complex example to show the consequence of a speed constraint on aircraft converging with a small angle. When the speed constraint is relaxed, the algorithm finds a solution, but it fails when no speed change is allowed. We have also shown that a centralized approach that coordinates maneuvers can handle the complex case. These findings strongly suggest that Air Traffic Management needs centralized coordination because of the constrained speed environment.

Building reliable resolution tools for solving conflicts is very challenging. Van den Berg et al.'s algorithm is very efficient in a non constrained context. For now, it modifies speed vectors of both aircraft by "sharing efforts" to reach the no conflict area. This rule is simple and does not require any coordination between aircraft. With a coordination process, it could be modified to increase chances of solving more complex conflicts. This needs to be investigated.

The centralized approach used in section III needs to be tested on the scenarios used for measuring the autonomous approach, and not only on the complex case presented. This would help show the advantage of keeping centralized control of air traffic more generally. Results could be used as a reference for comparing with further improvements of this autonomous approach or with other autonomous approaches.

REFERENCES

[1] K. Zeghal, "A comparison of different approaches based on force fields for coordination among multiple mobiles," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, pp. 273–278 vol.1, Oct 1998.

[2] J. Kosecka, C. Tomlin, G. Pappas, and S. Sastry, "2-1/2 d conflict resolution maneuvers for atms," in *in Proc. 37th IEEE Conf. Decision Control*, pp. 2650–2655, 1998.

[3] M. Eby and I. Kelly, W.E., "Free flight separation assurance using distributed algorithms," in *Aerospace Conference, 1999. Proceedings. 1999 IEEE*, vol. 2, pp. 429–441 vol.2, 1999.

[4] G. Granger, N. Durand, and J. Alliot, "Token allocation strategy for free-flight conflict solving," in *IJCAI'01*, 2001.

[5] G. Granger, N. Durand, and J. Alliot, "Optimal resolution of en route conflicts," in *4th ATM R and D Seminar*, 2001.

[6] N. Archambault and N. Durand, "Scheduling heuristics for on-board sequential air conflict solving," in *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, vol. 1, pp. –3.1–9 Vol.1, Oct 2004.

[7] N. Durand, J. Alliot, and F. Medioni, "Neural nets trained by genetic algorithms for collision avoidance," *Applied Artificial Intelligence*, 2000.

[8] I. Hwang, J.Kim, and C. Tomlin, "Protocol-based conflict resolution for air traffic control," *ATC Quarterly*, vol. 15, pp. 1–34, 2007.

[9] J. Le Ny and G. Pappas, "Geometric programming and mechanism design for air traffic conflict resolution," in *American Control Conference (ACC), 2010*, pp. 3069–3074, June 2010.

[10] J. Snape and D. Manocha, "Navigating multiple simple-airplanes in 3d workspace," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3974–3980, 2010.

[11] J. van den Berg, S. J. Guy, M. C. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *14th International Symposium on Robotics Research*, pp. 3–19, 2011.

[12] N. Durand, J.-M. Alliot, and J. Noailles, "Automatic aircraft conflict resolution using genetic algorithms," in *Proceedings of the Symposium on Applied Computing, Philadelphia*, ACM, 1996.

[13] N. Durand and J.-M. Alliot, "Optimal resolution of en route conflicts," in *First ATM Seminar Europe/USA, Saclay*, 1997.

[14] F. Krella *et al.*, "Arc 2000 scenario (version 4.3)," tech. rep., Eurocontrol, April 1989.

[15] Y.-J. Chiang, J. Klosowski, C. Lee, and J. Mitchell, "Geometric algorithms for conflict detection/resolution in air traffic management," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 2, pp. 1835–1840 vol.2, Dec 1997.

[16] J. Hu, M. Prandini, A. Nilim, and S. Sastry, "Optimal coordinated maneuvers for threedimensional aircraft conflict resolution," *AIAA Journal of Guidance, Control and Dynamics*, vol. 25, p. 2002, 2002.

[17] J.-H. Oh, J. Shewchun, and E. Feron, "Design and analysis of conflict resolution algorithms via positive semidefinite programming [aircraft conflict resolution]," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 5, pp. 4179–4185 vol.5, Dec 1997.

[18] E. Frazzoli, Z.-H. Mao, J.-H. Oh, and E. Feron, "Resolution of conflicts involving many aircraft via semidefinite programming," *AIAA Journal of Guidance, Control and Dynamics*, vol. 24, Jan-Feb 2001.

[19] L. Pallottino, A. Bicchi, and E. Feron, "Mixed integer programming for aircraft conflict resolution," in *AIAA Guidance Navigation and Control Conference and Exhibit*, 2001.

[20] L. Pallottino, E. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, pp. 3–11, Mar 2002.

[21] M. A. Christodoulou and C. Kontogeorgou, "Collision avoidance in commercial aircraft free flight via neural networks and non-linear programming," *Int. J. Neural Syst.*, vol. 18, no. 5, pp. 371–387, 2008.

[22] M. Gariel and E. Feron, "3d conflict avoidance under uncertainties," in *Digital Avionics Systems Conference, 2009. DASC '09. IEEE/AIAA 28th*, pp. 4.E.3–1–4.E.3–8, Oct 2009.

[23] C. Allignol, N. Barnier, N. Durand, and J.-M. Alliot, "A new framework for solving en-routes conflicts," in *10th USA/Europe Air Traffic Management Research and Developpment Seminar*, 2013.

[24] P. Averty, B. Johansson, J. Wise, and C. Caspie, "Could erasmus speed adjustments be identifiable by air traffic controllers ?," in *7th ATM R and D Seminar*, 2007.

[25] F. Drogoul, P. Averty, and R. Weber, "Erasmus strategic deconfliction to benefit sesar," in *8th ATM R and D Seminar*, 2009.

Nicolas Durand graduated from the École Polytechnique de Paris in 1990 and the École Nationale de l'Aviation Civile (ENAC) in 1992. He has been a design engineer at the Centre d'Études de la Navigation Aérienne (now DSNA/DTI R&D) since 1992, holds a Ph.D. in Computer Science (1996) and got his HDR (french equivalent of tenure) in 2004. He is currently professor at the ENAC/MAIAA (Applied Maths, Computer Science, Automatic applied to Aviation) laboratory.

Nicolas Barnier is a lecturer and research assistant at ENAC. He graduated from ENAC as an engineer in 1997, and received a Ph.D. in computer science in 2002 from the University of Toulouse. He is one of the authors of FaCiLe, an open source Constraint Programming library for the functional language OCaml. His research interests focus on Constraint Programming, Local Search and Combinatorial Optimization in general.