Token Allocation Strategy for Free-Flight Conflict Solving

Track: Emerging Applications, Technology and Issues

Géraud Granger and **Nicolas Durand** and **Jean-Marc Alliot**

CENA*/LOG†

7, av Ed Belin

31055 Toulouse Cedex France

tel: (33) 5 62 17 40 54 - fax: (33) 5 62 17 41 43

email: granger@recherche.enac.fr - durand@tls.cena.fr - alliot@dgac.fr

Abstract

For the last 10 years, airlines have widely supported research on the development of airspaces where aircraft would be free to decide their trajectory: these areas where called Free-flight airspaces. However, as soon as two aircraft are in the same area, their separation must be guaranteed. FACES[1] is an autonomous and coordinated embarked (on board) conflict solver for Free Flight airspace. It solves conflict by computing simple manoeuvres that guarantees conflict free trajectories for the next 5 minutes (min). Coordination is ensured by giving sequential manoeuvres to aircraft with a token allocation strategy. FACES can be implemented with the current positioning, broadcasting and flight management technology. Moreover, it is robust to communication or system failure for time up to one or two minutes. FACES was tested with a traffic simulator on busy traffic days over France. Airspace over flight level[2] 320 was considered as Free Flight.

Introduction

We have all experienced at least once a long wait in an overcrowded air terminal. Reading magazines distributed by airlines during these long hours, we often found that they consider air traffic control as one of the major cause for delays. And it is true that the air traffic control system is becoming saturated. But, if delays due to overloaded airports are easy to understand, it is much harder to comprehend delays due to the En Route control system. In fact, if we ask a mathematician to analyze the system in cold blood, it can be proved that the collision probability over flight level 320 is very low for aircraft flying direct routes, especially if some elementary precautions are taken regarding face to face or overtaking conflicts. So, En Route control could be considered as expensive (En route charges), inefficient (delays induced) and statistically of very little use. However, if the Free Route and Free Flight concepts are attractive, especially to airlines, we still must consider safety as the first priority, and design new algorithms and systems for these new airspaces (airspace above flight level 320).

---

*Centre d'Etudes de la Navigation Aérienne

†Laboratoire d'Optimisation Globale

[1]Free flight Autonomous and Coordinated Embarked Solver

[2]flight level X or FL X stands for $X \times 100$ feet of altitude

The most well known reactive collision avoidance concept is certainly the ACAS/TCAS system. It is a very short term collision avoidance system (less than 60 seconds) and should only be looked upon as the last safety filter of an ATC system. Experiments with TCAS to control aircraft on simulated traffic have shown that poor coordination could leads to disastrous situations (Bosc 1997). Other simple techniques using repulsive forces (Zeghal 1998b; 1998a) have also been investigated but drawbacks remain (Bosc 1997).

This paper presents an algorithm for autonomous embarked (on board) conflict resolution with a coordination mechanism, called FACES (Free-flight Autonomous Coordinated Embarked Solver). Moreover, it is robust to communication or system failure for time up to one or two minutes. This algorithm could be implemented with current technology (GPS, FMS, ADS-B) at low cost.

Section deals with the hypothesis we made and the modelling chosen. Section presents the ordering strategy. In section , the A^* algorithm used to optimize a conflict free trajectory for one aircraft is detailed. In section , the basic algorithm is tested with the air traffic simulator CATS (Alliot *et al.* 1997) on a heavily loaded traffic day in the French airspace over flight level[1] 320. Experimental results led us to introduce slight improvements to the basic algorithm also presented in section ; then we finally discuss results of simulations on the enhanced algorithm.

Modelling

Hypothesis

The idea is to build an embarked - that is, an on-board - solver able to compute a manoeuvre each time a conflict is detected with another aircraft in a defined detection area around the aircraft. This solver should continuously (every minute[1]) guarantee a 5^3 minutes conflict free trajectory to each aircraft. This 5 minutes conflict free period guarantees that a transient failure of communications would not have a disastrous effect: the system could still restart later on; resolutions would be less optimal, more vertical manoeuvres could be necessary to solve all conflicts, as anticipation would be shorter, but the risk of collision would remain close to zero.

---

[3]These parameters can be modified. This first study does not discuss the opportunity of increasing or decreasing these values.

Manoeuvres suggested have to be simple to understand and to execute. No manoeuvre can be given during the first minute (called the quiescent period) in order to give enough time to the solver to compute a solution and inform the pilot (or directly program the FMS). Moreover, only one manoeuvre can be given to one aircraft during a 5 minutes time window, and no manoeuvre can start as long as the previous one is not finished.

The algorithm enforces a global resolution order between conflicting aircraft. The general principle is as follows: the aircraft which is first chooses its trajectory without considering other aircraft. Then, the next aircraft in the priority queue takes this trajectory into account, and computes its own, and so on (see section).

Airspace over flight level[1] 320 is considered as a Free Flight airspace. This area is not a so low density area, especially in France. So it is an excellent test zone for a Free Flight solver. All aircraft entering this airspace are supposed to be separated for 5 minutes when entering the Free Flight zone, and are sent back separated for the next 5 minutes when leaving it. All aircraft entering this airspace have to be Free Flight compliant, i.e:

- they all have synchronous clock;

- they are able to receive all broadcast information from other aircraft which are within a 90 nmi zone around them (see part);

- they are all equipped with the FACES solver.

- they are able each minute at the same time to compute, and store their current position, their Free Flight airspace exit point and their predicted trajectory for the next 5 minutes.

- they are able to reliably broadcast the latter information as soon as it has been computed.

This information consists of 20 3D-points, one every 15 seconds (in fact, only 16 are needed, those beginning at $t = 1$ min). Extra information is added to the predicted position that indicates its accuracy (the uncertainty model is detailed in part). Of course, the more accurate the information, the more efficient detection and resolution. This prediction has to be *contractual*, i.e. as soon as an aircraft has broadcast these informations, it has to keep to this trajectory for the next 5 minutes as long as the solver does not give a manoeuvre. It must be noticed that on exceptional occasions, one aircraft can modify this trajectory, or aircraft not equipped for Free Flight can be accepted in the Free Flight zone. This can also take into account exceptional events such as the failure of one aircraft conflict solver. Theses aircraft will be given the highest priority number (see part) and all other aircraft will build their trajectory in order to avoid them. This should be a last resort as the algorithm might fail if two such aircraft are present at the same time in the same zone.

Manoeuvre modelling

As stated above, time is discretized into 15 seconds[4] time steps. As manoeuvres must remain simple to understand and execute, the turning point modelling is chosen in the horizontal plane (see figure 1). In this article, no manoeuvre is given in the vertical plane[5]. As shown on figure 1, a ma-
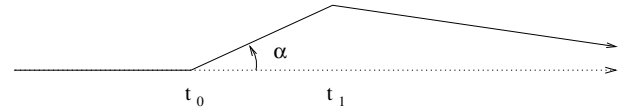

Figure 1: Turning point modelling.

noeuvre is a heading change of 10, 20 and 30 degrees right or left, it starts at time t_0, and ends at time t_1. As stated above, t_0 (and t_1) are always larger than 1 minute.

Uncertainty modelling and 1-to-1 conflict detection

A very simple filter is first applied: only aircraft within a 90 nmi zone are considered as being potential threats. This radius is such that 2 aircraft facing each other at 500 kn cannot be in conflict[6] during the next 5 minutes if they are not in the detection zone of the facing aircraft.

We then assume that there is an error about the aircraft's future location because of ground speed prediction uncertainties[7]. Uncertainties on climbing and descending rates are even more important[8]. Uncertainties on the future positions of aircraft are all the more important because the prediction is faraway.

In the vertical plane, we use a cylindrical modelling (figure 2). Each aircraft has a maximal altitude and a minimal altitude. To check if two aircraft are in conflict, the minimal altitude of the higher aircraft is compared to the maximal altitude of the lower aircraft.

In the horizontal plane, an aircraft is represented by a point at the initial time. The point becomes a line segment in the uncertainty direction (the speed direction here, see figure 2). The first point of the line "flies" at the maximum possible speed, and the last point at the minimum possible speed. When changing direction ($t = 4$), the segment becomes a parallelogram that increases in the speed direction. When changing a second time direction ($t = 7$), the parallelogram

---

[4]This value is not chosen at random. With 15 s time steps, detection can be made only on these points (and not on the segments between these points) with the guaranty that two aircraft can not cross each other without noticing a serious conflict.

[5]Vertical manoeuvres were put aside on purpose. They are more difficult to execute, and less comfortable for both pilots and passengers. Results of part show that they should only be used as a last resort, on the very rare occasions where the solver fails.

[6]In this article the separation standards are 6 nmi in the horizontal plane and 1000 ft in the vertical plane.

[7]Uncertainties on ground track will not be considered, as they do not increase with time and will be included in the separation standard.

[8]The error percentages on vertical and horizontal speed are specific to each aircraft. For example, aircraft with very accurate FMS will have very low percentages.
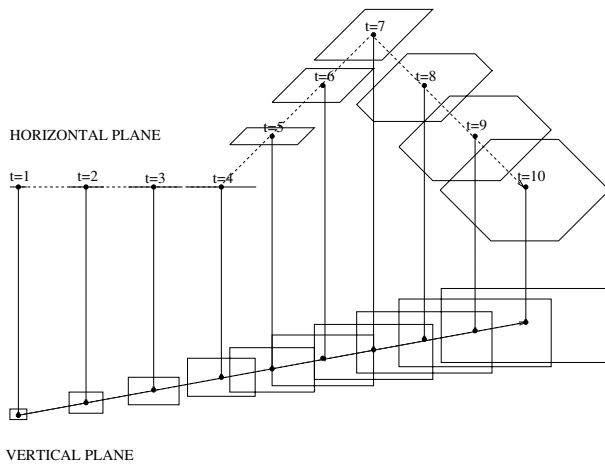
Figure 2: Modelling of speed uncertainties.

becomes a hexagon that increases in the new speed direction. To check the separation standard at time t, we compute the distance between the two polygons modelling the aircraft positions and compare it to the separation standard at each time step of the simulation. It must be noticed that, as only one manoeuvre can be given in a 5 minutes time window, and as no manoeuvre can start as long as the previous one is not finished, the convex can only be a line, a parallelogram or an hexagon.

A classical problem in 1 to 1 conflict detection is symmetry. If aircraft A considers it is in conflict with aircraft B, then B must consider A as a conflicting aircraft. In FACES, broadcasting of positions guarantees that two aircraft that can detect each other share exactly the same information regarding their positions. As detection algorithms are identical, 1 to 1 detection will always be symmetrical.

Ordering strategy

The coordination problem

Centralized automatic solvers as described by N. Durand(Durand 1996) find a global solution to clusters involving many aircraft. Manoeuvres are then given to aircraft simultaneously. An on board solver cannot be based on the same principle: aircraft do not share the same information, as they do not have the same detection zone (limited to 90nmi). A coordination problem appears and must be solved.

The Free-R (VU N. Duong 1997) project uses extended flight rules to solve this problem. The TCAS system uses the transponder code to decide which aircraft has to manoeuvre; giving resolution priorities to aircraft is a way often adopted for solving the coordination problem.

A resolution priority order[9] has to be total if we want each aircraft to solve all conflicts when there is more than 2 aircraft. For example, the Visual Flight Rule that gives priority to the aircraft coming from the right does not define a global

---

[9]An order relation must be anti-symmetrical and transitive. An order relation is total if every pair of individuals can be compared.
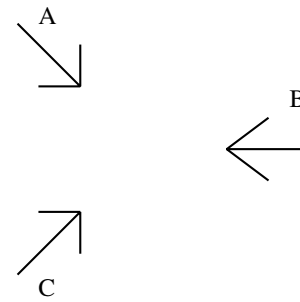

Figure 3: 3 aircraft conflict.

order if there are more than 2 aircraft simultaneously in conflict. Figure 3 gives an example of conflict involving 3 aircraft for which this priority resolution rule does not define an order because transitivity is not ensured.

Moreover, as aircraft do not share the same information, defining which aircraft will be the first to choose its trajectory is not obvious, even if we have a total priority order.

Building a global resolution order

A token allocation strategy   We have to define a global resolution order such as each aircraft can know when it can start to build a conflict free trajectory and which aircraft it has to avoid.

We first suppose that a total priority order relation exists on the aircraft population. We can use the simple order based on transponder numbers discussed above (but more elaborated orders will be discussed in part). At each resolution step, we build a global resolution order from this priority order with the following strategy:

1. First, every aircraft sends its predicted trajectory to its neighbours. Each aircraft is then able to know wether it is conflicting with another aircraft or not for the next five minutes.

2. Each aircraft receives a token from every conflicting aircraft which has a higher priority in its detection zone. Aircraft that are not in conflict never receive any token.

3. Then, each conflicting aircraft with no token solves conflicts with every aircraft in its detection zone that has no token (at the first iteration step, they will just go straight). It does not take into account aircraft that have one or more tokens.

4. When this trajectory has been computed, the aircraft broadcasts its new trajectory; all aircraft which have received a token from this aircraft take this new trajectory into account, and cancel the token received from this aircraft.

5. Steps 3 and 4 are repeated until no token remains.

Detailed example   The following example has been observed in the simulations presented in part . 8 aircraft belong to the same cluster (figure 4). The detection area of each aircraft is given on the figure.

Aircraft $A7$ detects every other aircraft. $A8$ detects $A5$, $A6$ and $A7$. $A4$ detects $A1$, $A2$ and $A7$. $A6$ detects $A2$, $A3$,

Step	1	2	3	4	5	6
A7	6	4	3	1	0	
A8	4	3	3	2	1	0
A4	2	1	0			
A6	2	2	1	0		
A2	1	0				
A3	1	1	0			
A1	0					
A5	0					

Table 1: Token allocation at the different steps of resolution

$A7$ and $A8$. $A2$ detects $A1$, $A3$, $A4$, $A6$ and $A7$. $A3$ detects $A2$, $A6$ and $A7$. $A1$ detects $A2$, $A4$ and $A7$. $A5$ detects $A7$ and $A8$. Conflicting aircraft are $A1 - A2$, $A1 - A4$, $A2 - A3$, $A2 - A7$, $A3 - A6$, $A5 - A7$ and $A5 - A8$. Aircraft that has the highest priority is $A1$ and the lowest priority order is $A8$ ($A1 > A2 > A3 > A4 > A5 > A6 > A7 > A8$).

− is in conflict with
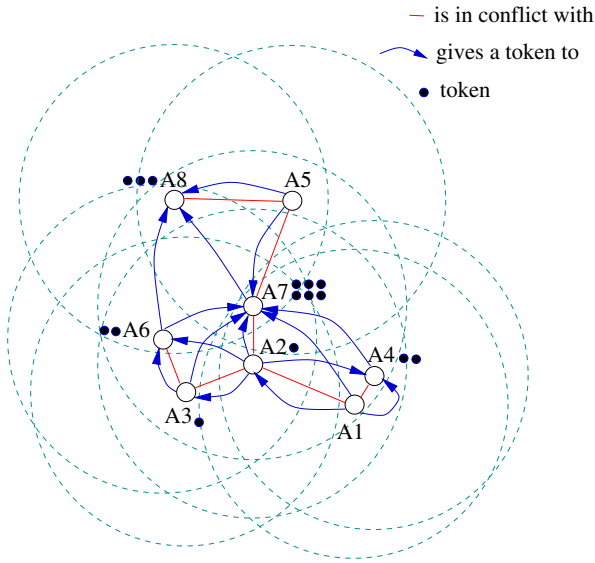⟶ gives a token to
• token

Figure 4: Cluster of aircraft in conflict

Tokens are allocated as presented on figure 4.

Table 1 gives the token allocation at the different steps of the resolution.

During step 1, $A1$ and $A5$ (0 token) choose their trajectories without considering other aircraft in their detection area (which have at least 1 token). Then, they broadcast their unmodified trajectories and all aircraft that have received a token from them cancel it. $A2$, $A4$ and $A7$ cancel the tokens sent by $A1$, $A7$ and $A8$ cancel the tokens sent by $A5$. During step 2, $A2$ (0 token) has no token and modifies its trajectory to solve conflict with $A1$ (0 token). $A3$, $A4$, $A6$, and $A7$ cancel the tokens sent by $A2$. During step 3, $A3$ (0 token) modifies its trajectory to solve conflict with $A2$ (0 token), $A4$ (0 token) modifies its trajectory to solve conflict with $A1$ (0 token) ;the new trajectory must not interfere with $A2$ (0 token). $A6$ cancels one token sent by $A3$ and $A7$ cancels

two tokens sent by $A3$ and $A4$. During step 4, $A6$ (0 token) modifies its trajectory to solve conflict with $A3$ (0 token), the new trajectory must not interfere with $A2$ (0 token). $A7$ and $A8$ cancel one token sent by $A6$. During step 5, $A7$ (0 token) modifies its trajectory to solve conflict with $A2$ (0 token) and $A5$ (0 token); the new trajectory must not interfere with $A1$ (0 token), $A3$ (0 token), $A4$ (0 token) and $A6$ (0 token). $A8$ cancels the token sent by $A7$. 1. During step 6, $A8$ (0 token) modifies its trajectory to solve conflict with $A5$ (0 token); the new trajectory must not interfere with $A6$ (0 token) and $A7$ (0 token).

Provability

The allocation-resolution method described above cannot lead to situations where all aircraft would have at least one token or situations where two aircraft detecting each other without any token would have to solve simultaneously. This is guaranteed by the use of a total priority order on aircraft. At each step, an aircraft with no token cannot have any other conflicting aircraft (that has not already solved) with no token in its detection area. In such a case, one of these two aircraft would have given a token to the other. At each step, among the conflicting aircraft that have not already solved, there is one that has the highest priority. This aircraft cannot have any token. It can solve and get back its tokens. The algorithm can be mathematically proved.

The A^* algorithm

As soon as the resolution order is chosen, the problem is to solve a 1 to n conflict problem: we have to find the minimum length trajectory for an aircraft avoiding n already fixed aircraft trajectories, that can be considered as obstacles. This is a classical robotics problem, therefore a classical A^* algorithm (see (Pearl 1984)) is used.

In the present application, the initial state is the state of the solving aircraft at $t = 1$ minute. The terminal states are the possible states of the solving aircraft after 5 minutes of flight or when they have reached their destination.

Each branch of the tree represents a possible trajectory of the solving aircraft. Fortunately, the heuristic function is used to only develop a small part of the tree.

The cost of a path is the trajectory length described by this path. Before starting a manoeuvre, an aircraft is in S_0 state. At each time step, each S_0 state generates 6 S_1 states corresponding to the 6 possible deviations of the trajectory (10, 20, 30 degrees right or left), and 1 S_0 state (the aircraft is not manoeuvred). At each time step, each S_1 state generates one S_1 state (the manoeuvre is extended) and one S_2 state (the aircraft is sent back to its Free Flight zone exit point). Every state generates a terminal S_3 state after 5 minutes or if the aircraft has reached its destination.

The cost function $k(u, v)$ measures the distance between the position of the aircraft at node u (time step t_{t_u}, state S_{s_u}) and the position of the aircraft at node v (time step t_{t_v}, state S_{s_v}). If a conflict occurs between node u and node v, the value $k(u, v)$ is widely increased so that the corresponding branch is no longer developed.

The heuristic function $h(u)$ is here the direct distance between node u and the Free Flight exit point (destination) of

the solving aircraft. This heuristic is clearly an underestimating one, which guarantees that the optimal solution will always be found.

Generally, many different paths are developed and the depth of the tree is 16 (4 minutes). In this applications, the solution is given in less than 5 seconds on a Pentium II 300, even for the biggest 1-to-35 conflict.

Experimental results and improvements

The CATS simulator

The algorithm described in part and was tested on the CATS (Alliot *et al.* 1997) simulator. The core of the CATS system is an En-route traffic simulation engine. It is based on a discrete, fixed time slice execution model: the position and speed of aircraft are computed at fixed time steps, usually every 5, 10 or 15 seconds.

Aircraft performances are in tabulated form describing ground speed, vertical speed, and fuel burn as a function of altitude, aircraft type and flight segment (cruise, climb or descent.)

In the further applications, aircraft use direct routes to their destination. The separation standard used is 6 nautical miles in the horizontal plane and 1000 ft vertically[10]. Conflicts were not solved under flight level 320, and a delay t_e was added when necessary for aircraft entering the Free Flight zone in order to separate them on entry points. Uncertainties on speed (either vertical or horizontal) were set to minimal values.

Results

Results presented in this part are obtained with the 6381 flight plans of the 21st of June 1996 with no regulation. The Free Flight zone defined is the airspace above flight level 320. The allocation-resolution strategy described in part is repeated every minute and the trajectory prediction is done on the next 5 minutes.

It was found that 2763 aircraft enter the Free Flight zone. 641 conflicts are detected in this zone during the day.

As described in section , the algorithm requires the definition of a total order among aircraft.

The following order is used: an aircraft that is manoeuvre free has a lower priority order than an aircraft that has already started a manoeuvre. The CAUTRA number is used to compare two manoeuvre free aircraft or two manoeuvred aircraft. A maneuver efficiency criteria is also added to the cost criteria in order to prevent aircraft from postponing a crossing maneuver when necessary. Therefore, when two aircraft must cross, the manoeuvre that enforces crossing must have a lower cost than the manoeuvre that postpones the crossing. This new criteria is included in the A^* algorithm.

With this new priority order, the A^* algorithm is called 2654 times. At the end of the simulation, 3 conflicts remain

---

[10]The 6 nmi value is certainly quite high for an on-board solver. We do think that it could seriously be reduced, regarding GPS and FMS precision. This would even improve airspace capacity and resolution efficiency.

unsolved, but they can easily be solved by a very simple vertical manoeuvre.

Delays

There were 897 manoeuvres given during the day to 367 aircraft (2.44 manoeuvre per aircraft). 13.28% of the aircraft flying in the Free Flight zone are manoeuvred.

73 aircraft are delayed when entering the Free Flight zone, because they are not conflict-free at that moment (that is 2.64% of the traffic in the Free Flight zone).

Delays are given in table 2.

Delay	Mean per aircraft	Mean per delayed acft	Max
while entering	$2s$	$1mn14s$	$3mn$
inside (man)	$3,6s$	$27,4s$	$40s$

Table 2: Delays

The maximum manoeuvre lasts $2 : 30$ minutes.

Table 3 gives the number of steps aircraft have to wait because they have been given tokens. In the most complex conflict, one aircraft has to wait for 7 resolutions before it can choose its trajectory.

Number of steps	Waiting aircraft	percentage / Total
1	1516	$76,6\%$
2	253	$12,78\%$
3	162	$8,19\%$
4	27	$1,36\%$
5	17	$0,86\%$
6	3	$0,16\%$
7	1	$0,05\%$

Table 3: Number of waiting aircraft.

So, regarding delays, the performance of the algorithm is very good.

Unsolved conflicts and priority order

There are 3 aircraft in each remaining unsolved conflicts. These conflicts appear because the order between aircraft is not well chosen.

Moreover, it looks extremely difficult to devise an algorithm that would find the best possible order without seriously increasing the complexity of the global algorithm and the necessary capacity of the communication medium. On-board conflict solvers which have only a partial information on the global situation will almost certainly remain suboptimal, while centralized conflict solvers are able to find the global optimal solutions. However, this may not be a too serious concern in the upper airspace: the simulation above shows that this algorithm is almost always able to solve conflicts, even with situations as complex as the one presented on on figure 5 where 35 aircraft are involved, while delays remain small.
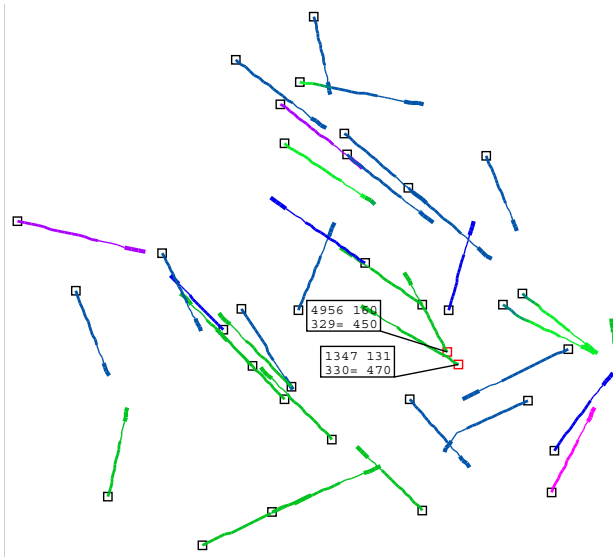
Figure 5: 35 aircraft in the detection zone.

Conclusion

We have demonstrated in this article that an efficient on board algorithm for Free Flight conflict resolution can be designed and implemented. This algorithm has the following advantages:

- Compared to a centralized automated system, the development of such a system could be relatively low cost. Most hypothesis are quite weak: synchronous clocks are already available with GPS, FMS are now elaborate enough to provide the information needed for trajectory prediction in the next 5 minutes, provides sufficient capacity for communications; the 1 to n resolution algorithm is simple to implement and has already been widely used for similar problems in robotics; computing power needed fits in a standard PC computer.

- Compared to rule based system, the algorithm is mathematically provable, and the simulation above shows that it would be efficient in upper airspace, even when density is quite high

- Compared to purely reactive systems (Zeghal 1998b), which usually requires permanent changes in headings, the manoeuvre model is classical and easy to implement. Further, and this is the main point to stress, as trajectories are guaranteed conflict free for at least 5 minutes, a transient failure of communications would not have a disastrous effect: the system could still restart later on; resolutions would be less optimal, more vertical manoeuvres could be necessary to solve all conflicts, as anticipation would be shorter, but the risk of collision would remain insignificant.

- The system could be progressively put into service by first defining Free Flight airspace over oceanic areas, and gradually extending them. This would help solving the classical transition problem from the current system to a partially automated one.

We are aware that the whole system depends on the fiability and availability of transmissions. Requirements on the bandwidth are low enough to enable multiple emissions of messages. But error correlations would have to be considered. We miss informations and results on these issues. However, we believe that an airborne implementation of this algorithm can be seriously considered.

References

Alliot, J.; Durand, N.; Bosc, J.; and Maugis, L. 1997. Cats: a complete air traffic simulator. *16th AIAA/IEEE Digital Avionics Systems Conference, IRVINE*.

Bosc, J.-F. 1997. *Techniques d'évitement réactif et simulation du trafic aérien*. Ph.D. Dissertation, Ecole Nationale de L'aviation civile.

Durand, N. 1996. *Optimisation de Trajectoires pour la Résolution de Conflits en Route*. Ph.D. Dissertation, EN-SEEIHT, Institut National Polytechnique de Toulouse.

Granger, G.; Durand, N.; and Alliot, J. 1998. FACES: a Free flight Autonomous and Coordinated Embarked Solver. In *2 ST U.S.A/EUROPE ATM R & D Seminar*.

Pearl, J. 1984. *Heuristics*. Addison-Wesley.

VU N. Duong, Eric Hoffman, J.-P. N. 1997. Autonomous aircraft. *1st U.S.A. / EUROPE Air Traffic Management R&D Seminar, SACLAY*.

Zeghal, K. 1998a. A comparison of different approaches based on force fields for coordination among multiple mobiles. In *IEEE International Conference on Intelligent Robotic System (IROS)*.

Zeghal, K. 1998b. A review of different approaches based on force fields for airborne conflict resolution. *AIAA Guidance, Navigation and Control Conference*.