The Influence of Uncertainties on Traffic Control using Speed Adjustments

Géraud Granger
STERIA
7, av Edouard Belin
31055 Toulouse cedex France
granger@recherche.enac.fr

Cyril Allignol Nicolas Durand
Direction de la Technique et de l'Innovation/Domaine R&D
Direction des Services de la Navigation Arienne
7, av Edouard Belin
31055 Toulouse cedex France
name@tls.cena.fr

***Abstract*—The RTA (Required Time Arrival) capabilities of aircraft FMS (Flight Management Systems) offer new opportunities to solve mid-term horizon conflicts (20 minutes in advance) with small speed adjustments. The ERASMUS project has shown promising results of up to 80% conflict resolution using small speed adjustments in the $[-6\%;+3\%]$ range with 20 minutes advance notice. The hypotheses were based on very accurate trajectory predictions (TPs). In this article we show how the quality of these results decreases as the uncertainties on the trajectory prediction increase. Therefore we used the CATS (Complete Air Traffic Simulator) developed in the late 90s at CENA (Centre d'Etudes de la Navigation Aérienne) and constrained the solver to use only speed maneuvers for leveled or descending aircraft with different hypotheses on speed ranges and speed uncertainties. Results show that Traffic Control using Speed Adjustments (TCSA) can solve most of the conflicts even when we consider uncertainty on the TP. However, the number of maneuvers that need to be given to aircraft is highly influenced by the uncertainties used in the TP.**

Keywords: speed adjustment, traffic complexity, conflict resolution

I. INTRODUCTION

Traffic Control using Speed Adjustments (TCSA) has been recently studied through the ERASMUS project. Several experiments were carried out in the Aix-en-Provence control center to test the ability of controllers to detect speed changes, check the compatibility of an automatic ground based speed adjustment tool with the controllers and pilots' working methods, and measure the potential benefit of the concept [BDG09], [GGG$^+$07], [CL07], [Ehr04].

From a theoretical point of view, conflict resolution using speed adjustments has been studied by Pallotino, Feron and Bicci [PFB02] and later by Vela, Solak, Singhose and Clarke [VSSC09] using mixed integer linear programming in a simplified context where aircraft are leveled and can all change their speeds at the same time. It as also been studied by Durand and Alliot in the mid 90s with a stochastic optimization approach [DAN96]. This approach was used in the ERCOS (En Route Conflict Optimized Solver) [GDA01] that is integrated in the CATS (Complete Air Traffic Simulator) fast time simulator.

The current Air Traffic Management system can be conceptually divided into four layers or filters, each with a decreasing time horizon relative to the moment of departure:

1) Strategic (several months), ASM (Air Space Management): design of routes, sectors and procedures
2) (Pre-)Tactical (a few days to a few hours), ATFM (Air Traffic Flow Management): control centers opening schedules define hourly capacities of each open sectors (or groups of sectors). To respect these capacity constraints, the CFMU (Central Flow Management Unit) computes and updates flow regulations and reroutings according to the posted flight plans and resulting workload excess.
3) Real time (5/10 minutes), tactical control: surveillance, coordination with adjacent centers, conflict resolution by various simple maneuvers (heading, flight level, speed) transmitted to the pilots.
4) Emergency (less than 5 min), safety nets: groundbased (Short Term Conflict Alert, Minimum Safety Altitude Warning) and airborne (Traffic Alert and Collision Avoidance System, Ground Proximity Warning System).

The power of TCSA discribed by [BDG09] is that it has the potential to be added between layer 2 and layer 3 without affecting them. Consequently, it could increase the safety of the whole system by adding a layer and also increase the capacity if the number of conflicts solved is high enough. The purpose of this paper is to evaluate the potential number of conflicts that could be solved using fast time simulations on real data, and measure the effect of uncertainties on the results obtained.

Airlines and aircraft manufacturers will agree to pay for new onboard systems if it can be proven that the money invested provides some benefit in terms of efficiency (for example reducing delays) and/or safety. Slightly modifying the speeds of aircraft to increase their separation without interfering with the existing control system could add a safety layer, but the question is to measure the efficiency of the concept for different scenarios of uncertainties.

In this paper we propose to analyse the influence of speed uncertainty on the efficiency of mid-term speed adjustments. Therefore we test different scenarios of speed adjustments using a [-5%,+5%] speed range, an anticipation time of 20 minutes corresponding to a mid-term time horizon, and two different update time steps (3 and 5) minutes. These scenarios are tested on real 2010 traffic data in the French airspace with different ground speed uncertainties (2%, 5% or 10%) and different vertical speed uncertainties (10% or 30%).

In the first part, we describe CATS, the simulator used. The second part details the maneuver modeling and the solver algorithm used. The third part describes the solver algorithm. In the last part, numerical results are presented.

II. THE COMPLETE AIR TRAFFIC SIMULATOR (CATS)

A. *General framework*

Results presented in this paper rely on real data traffic simulation. The CAT Simulator takes as input flight plans given by airlines or pilots with or without regulation. In the application, the traffic used is the French real traffic of July 17^{th} 2010. The simulator uses the BADA tabulated model for modeling aircraft performances. For a given aircraft type, it gives a vertical speed and a ground speed which depends on the aircraft attitude (whether it is climbing, leveled or descending). For example, a B747 leveled at FL-300 has a GS of 490 kts. If it is climbing, its GS will be 480 kts and its VS 1000 ft/mn. At FL-150, values would be respectively 430, 420 and 1800. 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). The main dataset for aircraft flight performance used is the base of aircraft data (BADA) performance summary tables derived from the total energy model of EUROCONTROL.

Aircraft speeds can be modified by a random value to take into account different factors of uncertainty (aircraft load, winds, etc…). These values can be either computed once at aircraft activation and remain the same for all the flight, or can be modified anytime during the flight. Uncertainty modeling for conflict detection and resolution is introduced in the next section.

Aircraft follow classical routes (from way-point to way-point). The flight model is simple. An aircraft first climbs up to its RFL (Requested Flight Level), then remains leveled until its top of descent, then descends to its destination.

Aircraft fly with a time step that can be chosen at the start of the simulation. The time step is always chosen in order to guarantee that two aircraft face to face flying at 500 kts could not cross without being closer than one standard separation at at least one time step. For all our simulations, we use a $15s$ time step.

B. *General architecture of the system*

We sketch here the architecture of the simulator. Each part will be detailed in the following sections. The system architecture is presented in figure 1 and 2. The system relies on three main processes P1, P2, and P3:

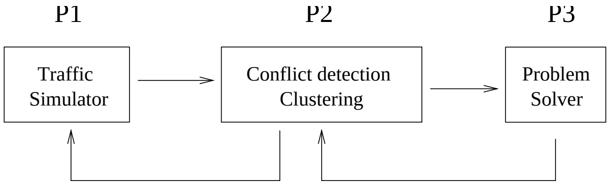

Fig. 1. General architecture

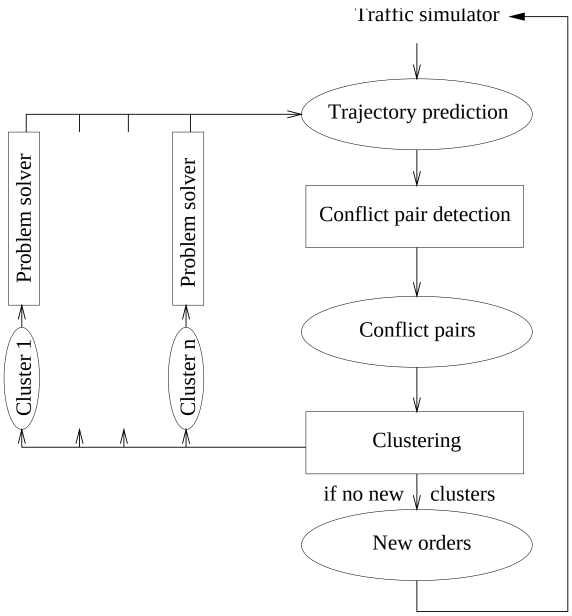

Fig. 2. Detailed architecture of the prototype

- P1 is the traffic simulator.
- P2 is in charge of conflict pair detection, clustering of pairs, and verification of new trajectories built by the solver.
- P3 is the problem solver.

P1 sends current aircraft positions and flight plans to process P2. Process P2 builds trajectory forecasts for T_w minutes, does conflict detection by pairs and transforms 1-to-1 conflicts in n-aircraft conflicts. Then, process P3 (the problem solver) solves in parallel each cluster, as Nicolas Durand graduated from the aircraft in each cluster are independent from aircraft in the other clusters. The problem solver sends to P2 new orders and P2 builds new trajectory forecasts based on these orders. Then P2 once again runs a conflict detection process to check that modified aircraft trajectories do not interfere with aircraft in another cluster, or with new aircraft. If no interference is found, new flight orders are sent to P1. If there are interferences, interfering clusters are joined and the solver is used again on that (these) cluster(s). The process is iterated until no interference between clusters remains, or no new aircraft is concerned by modified trajectories. The new orders are sent back to the traffic simulator.

The above process is iterated and all trajectories are optimized each δ minutes (3 or 5 minutes in the experiments). However, during the computation time, aircraft are flying and need to know if they must change their route or not. δ should be large enough to compute a solution, send it to the pilot and

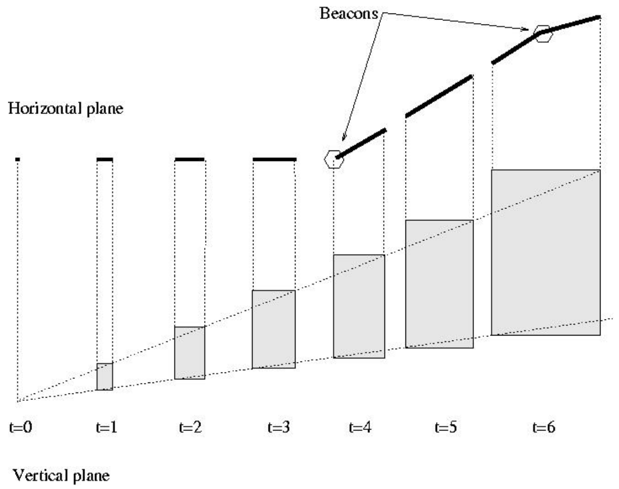

Fig. 3. Modeling of speed uncertainties.

leave him enough time to begin the maneuver. Consequently, for each aircraft, at the beginning of the current optimization, trajectories are determined by the previous run of the problem solver and cannot be changed for the next δ minutes.

Nicolas Durand graduated from the

C. Conflict detection and clustering

Trajectory forecast and 1-to-1 conflict detection

As described above, the P2 process does trajectory prediction for T_w minutes. This trajectory prediction is done again by a simulation on a slightly modified version of the Air Traffic simulator. But, as stated above, we assume that there is an error about the aircraft's future location because of ground speed prediction uncertainties[1]. Climbing and descending rate uncertainties are larger than ground speed uncertainties. Because the conflict free trajectory must be robust regarding these and many other uncertainties, an aircraft is represented by a point at the initial time. The point becomes a line segment in the direction of uncertainty (the speed direction here, see figure 3). The first point of the line "flies" at the maximum possible speed, and the last point at the minimum possible speed.

When changing direction on a beacon, the heading of the line segment's "fastest point" changes as described on figure 3.

To check the standard separation at time t, we compute the distance between the two line segments modeling the aircraft positions and compare it to the standard separation at each time step of the simulation.

In the vertical plane, we use a cylindrical modeling (figure 3). Each aircraft has a mean altitude, 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.

Let us take an example. A B747 is leaving its departing airport (altitude 0) at $t = 0$. Its climb rate is 1800 ft/mn and its ground speed is 175 kts. If we suppose that ground speed uncertainty is 5% and vertical speed uncertainty 20%, maximal and minimal climb rate are $1800 \times 1.2 = 2160$ft and $1800 \times 0.8 = 1440$ ft/mn and ground speeds are respectively 184 and 166 kts. This means that $15s$ later, the fastest and highest point has traveled 0.76 Nm and 540 ft while the slowest and lowest has only traveled 0.69 Nm and 360 ft. This time however, when computing maximal and minimal speeds, the difference of altitude of both points must be taken into account. At 540 ft, the tabulated model gives a standard ground speed of 197 kts, so max ground speed is $197 \times 1.2 = 237$ kts. At 360 ft, standard ground speed is 189 kts, with a minimal ground speed of 151 kts. Thus, the height of the segment grows much faster than the 20% factor for some aircraft.

Duration T_w can be changed, but must be at least equal to $2 \times \delta$. A good evaluation of T_w is difficult. With a perfect trajectory prediction, the largest T_w should be chosen. However, this is not true as soon as uncertainties are included in the model. A large value of T_w induces a large number of 1-to-1 conflicts, as the size of segments (modeling aircraft positions) grows quickly with time. Therefore, the conflict solver can become saturated.

Clustering

After pair detection, P2 clusters conflicting aircraft. Each equivalence class for the relation "is in conflict with" becomes a cluster.

For example, if aircraft A and B are in conflict in the T_w window, and if B is also in conflict with C in the same time window, then A, B, C is the same cluster and will be solved globally by the conflict solver.

The conflict solver sends back to P2 maneuver orders for solving conflicts. Then P2 computes new trajectories for all aircraft and checks if new interferences appear. For example, if the new trajectory given to aircraft B to solve the conflict with A and C interferes with cluster D, E and with aircraft F, then A, B, C, D, E, F will be sent back to the problem solver as one conflict to solve.

The process will always converge. In the worst case, P3 will have to solve a very large cluster including all aircraft present in the next T_w minutes. However, this technique is usually efficient because a very large number of clusters can be solved very quickly in parallel.

III. SOLVER MODELING

A. Maneuver modeling

In the CATS simulator, the conflict solver offers different maneuver options including heading changes (10, 20 or 30 degrees right or left), vertical maneuvers (flight level change, climbing interruption, descent anticipation, depending on if the aircraft is leveled, climbing or descending), or speed maneuvers. A maneuver starts at time t_1 and ends at time t_2. In this paper, we will limit the maneuvers to speed modifications when the aircraft is leveled or descending. We do not allow speed modifications during the climbing phase because many aircraft cannot change their speeds during this phase without changing the climbing rates. Different ranges ($[-15\%, +15\%]$,

---

[1] Ground track uncertainties will not be considered, as they do not increase with time and will be included in the standard separation

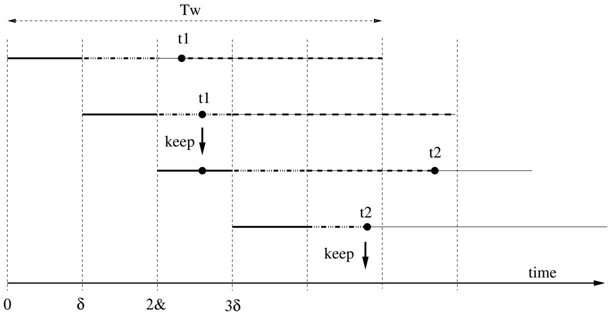

Fig. 4. The model and real time optimization.

$[-10\%, +10\%]$, $[-5\%, +5\%]$) of speed changes have been tested in the experiments in order to measure the influence of the this parameter.

A maneuver is determined by:

- t_0: the beginning of the speed change.
- α: the speed change
- t_1: the end of the speed change.

For a conflict involving n aircraft, the dimension of the search space is $3\,n$.

B. Maneuver decision time

Because of uncertainties, some conflicts that are detected too early would not actually occur in the end. Consequently, deciding to move aircraft in such cases would be useless and could even generate other conflicts that would not occur if no maneuver had been decided. This explains why controllers do not solve conflicts too early. When there is no uncertainty, the earlier the maneuver is started, the lower the delay. However, if speed is not strictly maintained, the earlier the conflict is detected, the lower the probability it will actually happen. Thus, a compromise must be reached between the delay generated and the risk of conflict.

Because of uncertainties, maneuvers should be started as late as possible with respect to the aircraft constraints. First, it prevents the system from deciding unnecessary maneuvers. Second, an aircraft that is already maneuvering (i.e. changing speed) cannot do another maneuver before resuming its initial speed. The solver was modeled this way to keep the maneuvers simple to understand and execute. Starting maneuvers as late as possible increases the number of maneuverable aircraft.

C. A sliding forecast time window model

In order to limit the size of the problem and to be reactive to uncertainties, only the next T_w minutes of the flights are considered. T_w represents the lookahead time also called *forecast time window*. The situtation is reconsidered every δ minutes with $\delta << T_w$. δ is the time step used in the model to make the T_w time window *slide*. This approach ensures that the problem can be updated every δ minutes : current aircraft positions are updated which reduces uncertainties.

In figure 4, at $t = 0$, the aircraft trajectory cannot be modified before $t = \delta$ because any maneuver requires advance notice. Any maneuver that would occur between $t = \delta$ and $t = 2\delta$ would be kept as a constraint for the next optimization run (in the example, no maneuver is decided). In figure 4, the maneuver described on the first line resulting from an optimization at $t = 0$ is reconsidered at time $t = \delta$ and then kept at time $t = 2\delta$. The end of the maneuver can be recalculated until the optimization starting at $t = 3\delta$.

Pilots should only be given maneuver orders that will not be modified; if no conflict occurs, no order will be given. In the example, the pilot will be notified of the beginning of the speed change at time δ and the end at time 3δ.

The size of the forecast time window is an important parameter. If it is too big, the size of the problem will include a very large number of variables and the resolution might be more difficult. If it is too small, the solutions found might be worse and the total delay induced over the day much higher. In order to solve conflicts with small speed adjustments, T_w needs to be large enough. Based on the ERASMUS experiments, we decided to set $T_w = 20min$ for this analysis.

IV. SOLVER ALGORITHM

Classical Evolutionary Computation (CEC) principles such as described in the literature [Gol89], [Mic92] is used in the solver.

A. Fitness function

The cost function used in this part is simply the sum of the delays over the aircraft population.

Solutions respecting the separation constraints cannot be built easily. Consequently, we need to include the separation constraint in the fitness function.

The fitness function chosen is:

$$F = \frac{n - \sum_{i=1}^{n} (\frac{\delta_i}{\delta_{max}})}{1 + nrc}$$

where n is the number of aircraft and nrc is the number of remaining conflicts.

The fitness function increases when the number of remaining conflicts and the delays decrease. It takes its values in $[0, n]$.

B. Crossover operator

The conflict resolution problem is partially separable as defined in [DA98], [DAN96]. In order to increase the probability of producing children with a better fitness than their parents, principles applied in [DA98] were used. For each aircraft i of a population element, a local fitness F_i value is defined as follows:

$$F_i = \frac{1 - (\frac{\delta_i}{\delta_{max}})}{1 + nrc_i}$$

where nrc_i is the number of remaining conflicts involving aircraft i.

Figure 5 presents the crossover operator. First two population elements are randomly chosen. For each parent A and B, fitness A_i and B_i of aircraft i are compared. If $A_i < B_i$, the children will take aircraft i of parent A. If $B_i < A_i$, the children will take aircraft i of parent B. If $A_i = B_i$ children randomly choose aircraft A_i or B_i or even a combination of A_i and B_i.

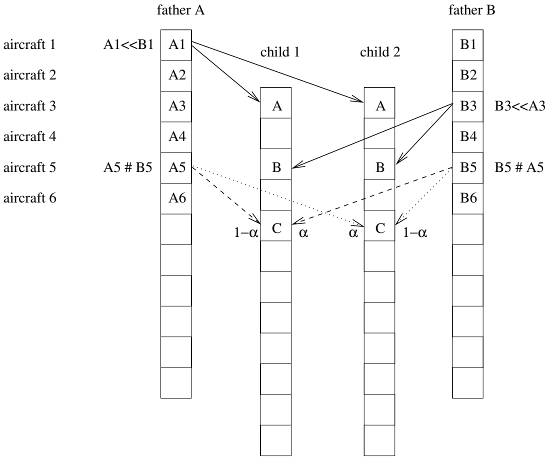

Fig. 5. Crossover operator

C. Mutation operator

For each candidate to mutation, the delay of an aircraft having one of the worst local fitnesses is modified. If every conflict is solved, an aircraft is randomly chosen and its parameters changed. In practice, a number m is randomly chosen in the interval $[1, \frac{n}{2}]$ and we pick up m times an aircraft to find the most constrained aircraft among these m trials. The delay of this aircraft is then either locally optimized or randomly modified with a probability of 50%. We may be tempted to always locally optimize the delay of the worst aircraft, but this would make the algorithm become very deterministic and lead to a premature convergence of the algorithm.

The crossover and mutation operators are more deterministic during the first generations because there are many conflicts to solve. They focus on making feasible solutions. When the solutions without conflicts appear in the population, they become less deterministic.

Sharing

The problem is highly combinatorial and may have many local optima. In order to prevent the algorithm from premature convergence, the sharing process introduced by Yin and Germay [YG93] is used. The complexity of this sharing process has the great advantage to be in $n\log(n)$ (instead of n^2 for classical sharing) if n is the size of the population. The distance used to compare two population elements p and q is:

$$D = \frac{\sum_{i=1}^{n} |\delta_i^p - \delta_i^q|}{n}$$

D. Parameters

In the experiments, the following parameters were empirically chosen: the size of the population was set to 100, 20% of the population is crossed, 60% is muted, the selection uses the *stochastic remainder without replacement*. A sharing process is used. As time to solve a problem is limited, the number of generations is limited to 500.

V. EXPERIMENTAL RESULTS

CATS was used on a busy day of traffic (July 17^{th} 2010) in the French airspace. We were only interested in the upper airspace (above FL195). The number of flights in this airspace is 8870 for that specific day. Without any control maneuver, 2305 conflicts are detected (26% of the aircraft population). The mean time of flights is 57 minutes and the mean travelled distance in the french airspace is 394 nautical miles. Different experiments were done using CATS with different hypotheses.

Tables I and II give the results of simulations for $\delta = 3mn$ and $\delta = 5mn$ for different values of horizontal uncertainty (0, 2, 5 or 10%[2]) and vertical uncertainty (10 or 30%). Columns 4 and 5 give the number and percentage of remaining conflicts. Column 4 also gives the number of remaining conflicts that involve two leveled aircraft. Columns 6 and 7 give the number of maneuvered aircraft and the percentage of aircraft maneuvered. Columns 7 and 8 give the number of maneuvers and mean number of maneuvers per aircraft.

A scenario with no uncertainty and a $[-15\%, +15\%]$ speed range was also tested (first line of tables I and II) in order to show what "theoretical" gain can be achieved on this specific day. In this case 97% of the conflicts could be solved by changing the speed of less than 2000 aircraft (20% of the aircraft population) which is very low. When the speed range is reduced to $[-5\%, +5\%]$ the number of remaining conflicts doubles with a number of aircraft maneuvered slightly above 2000. Those results show that with a perfect TP most of the conflicts could be solved 20 minutes in advance with a small speed adjustment. The lower lines of the tables show how the results worsen when the uncertainty on the TP increases. For example, with a 5% horizontal speed uncertainty and 10% vertical speed uncertainty the number of aircraft maneuvered to solve the conflicts is tripled and the number of remaining conflicts grows to 15% which is still good. Results show that vertical uncertainty increases the number of conflicts and maneuvers because many conflicts involve climbing or descending aircraft. In the worse case tested (10% of uncertainty on both horizontal and vertical speeds and a $\delta = 5mn$) the number of remaining conflicts reaches 50% and the number of necessary maneuvers increases dramatically.

5% of horizontal speed uncertainty corresponds to 20 knots. At this stage, we do not have the complete results for large uncertainties, but they will be available for the final version of this article. They should give us a more complete view of the importance of having an accurate TP on the ground to consider using TCSA.

When comparing the results using $\delta = 3mn$ to those using $\delta = 5mn$ it appears that the number of aircraft maneuvered increases by 10% and the number of remaining conflicts by 50%. This means that the solver is sensitive to the frequence of the detection and resolution cycle updates.

---

[2] Complete results for 10% of horizontal uncertainty will be available in the final version of this article

horiz uncert	vert uncert	speed range	remaining conf(leveled)	% of rem conflicts	maneuvered aircraft	% of acft maneuvered	nb of maneuvers	nb of man per aircraft
0	0	[-15 15]	78 (69)	3%	1889	21%	2397	0.27
0	0	[-10 10]	82 (72)	4%	1940	22%	2428	0.27
0	0	[-5 5]	144 (131)	6%	2102	24%	2605	0.29
2	10	[-5 5]	173 (142)	7%	3609	41%	5023	0.57
2	30	[-5 5]	191 (143)	8%	4663	53%	7093	0.80
5	10	[-5 5]	347 (244)	15%	5224	59%	8658	0.98
5	30	[-5 5]	392 (279)	17%	6194	70%	11447	1.29
10	10	[-5 5]	872 (612)	38%	6971	79%	14808	1.67
10	30	[-5 5]						

TABLE I

$\delta = 3mn$:NUMBER OF REMAINING CONFLICTS,NUMBER OF MANEUVERED AIRCRAFT AND NUMBER OF MANEUVERS FOR DIFFERENT UNCERTAINTIES AND DIFFERENT SPEED MODIFICATION RANGES.

horiz uncert	vert uncert	speed range	remaining conf(leveled)	% of rem conflicts	maneuvered aircraft	% of acft maneuvered	nb of maneuvers	nb of man per aircraft
0	0	[-15 15]	100 (84)	4%	2026	21%	2634	0.3
0	0	[-10 10]	107 (89)	5%	2057	23%	2700	0.3
0	0	[-5 5]	205 (166)	9%	2349	26%	3060	0.34
2	10	[-5 5]	251 (204)	11%	4162	47%	6058	0.68
2	30	[-5 5]	334 (232)	14%	5284	60%	8258	0.93
5	10	[-5 5]	517 (359)	22%	5773	65%	9688	1.09
5	30	[-5 5]	591 (386)	26%	6544	74%	11995	1.35
10	10	[-5 5]	1150 (801)	50%	7133	80%	14428	1.62
10	30	[-5 5]						

TABLE II

$\delta = 5mn$:NUMBER OF REMAINING CONFLICTS,NUMBER OF MANEUVERED AIRCRAFT AND NUMBER OF MANEUVERS FOR DIFFERENT UNCERTAINTIES AND DIFFERENT SPEED MODIFICATION RANGES.

Results also show that uncertainties have a big influence on the number of maneuvers needed to solve the conflicts. Controllers are used to saying that they often have to solve conflicts that might never happen because of uncertainties. It appears that an automatic TCSA tool would encounter the same issue.

The results obtained during the ERASMUS project were slightly different because the speed range used was $[-6\%, +3\%]$ and the seperation standard was 8 nautical miles.

VI. CONCLUSION

Controllers are aware of the fact that part of their task is to solve many conflicts that would never occur, because they have to deal with uncertainty.

Improvements in Trajectory Prediction offer new opportunities to enhance the efficiency and the safety of Air Traffic Management. Traffic Control using Speed Adjustments is one of the means that could be used to reach this goal. However, the expected gain of efficiency depends on the quality of the TP that can be used by the new tools.

This paper shows that having an accurate Trajectory Prediction is necessary to define a TCSA tool. These results do not suggest that we need to constrain the flight path to meet some Required Time Arrivals with a long advance notice. Instead, we need to be able to predict the future positions 20 minutes in advance with accuracy. Pilot attitude uncertainties are difficult to predict and our research was built on the strong hypothesis that the pilot would execute the maneuvers proposed by the solver using the Aircraft Flight Management System. Adding maneuver execution time uncertainty would probably worsen the results presented in this paper. This extra uncertainty needs to be studied carrefully. Having both controllers and pilots interacting with a medium term dolver is challenging because the controller will probably solve conflicts that are handled by the solver. This is why the ERASMUS experiments used a bigger separation standard. Pilot controller interaction with the medium term solver also needs to be modeled carefully and studied in future simulations to be able to give a realistic measure the possible gain of using TCSA.

REFERENCES

[BDG09] D. Bonini, C. Dupr, and G. Granger. How erasmus can support an increase in capacity in 2020. In *Proceedings of the 7th International Conference on Computing, Communications and Control Technologies: CCCT 2009, Orlando, Florida*, 2009.

[CL07] E. Cruck and J. Lygeros. Subliminal air traffic control: Human friendly control of a multi-agent system. In *American Control Conference*, 2007.

[DA98] Nicolas Durand and Jean-Marc Alliot. Genetic crossover operator for partially separable functions. In *Genetic Programming*, 1998.

[DAN96] Nicolas Durand, Jean-Marc Alliot, and Joseph Noailles. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the Symposium on Applied Computing, Philadelphia*. ACM, 1996.

[Ehr04] Ehrmanntraut. The potential of speed control. In *Digital Avionics Systems Conference*, 2004.

[GDA01] G. Granger, N. Durand, and J.M. Alliot. Optimal resolution of en route conflicts. In *4th ATM R and D Seminar*, 2001.

[GGG$^+$07] G. Gawinowski, J.-L. Garcia, R. Guerreau, R. Weber, , and M. Brochard. Erasmus: A new path for 4d trajecotry-based enablers to reduce the traffic complexity. In *Digital Avionics Systems Conference*, 2007.

[Gol89] D.E Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA Addison Wesley, 1989.

[Mic92] Z Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-verlag, 1992.

[PFB02] L. Pallottino, E. Feron, and A. Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems*, 3(1):3–11, 2002.

[VSSC09] A. Vela, S. Solak, W. Singhose, and J.P. Clarke. A mixed integer program for flight-level assignment and speed control for conflict resolution. In *Proceedings of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*. IEEE, 2009.

[YG93] Xiaodong Yin and Noel Germay. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In C.R. Reeves R.F.Albrecht and N.C. Steele, editors, *In proceedings of the Artificial Neural Nets and Genetic Algorithm International Conference, Insbruck Austria*. Springer-Verlag, 1993.

Géraud Granger graduated from the Ecole Nationale de l'Aviation Civile (ENAC) in 1998. He holds a Ph.D. in Computer Science from the Ecole Polytechnique de Paris (2002) and is currently a design engineer at STERIA working on SESAR.

Cyril Allignol graduated from the Ecole Nationale de l'Aviation Civile (ENAC) in 2006. He is currently a Ph.D. student at the DSNA/R&D Planning Optimization and Modeling team.

Nicolas Durand graduated from the Ecole polytechnique de Paris in 1990 and the Ecole Nationale de l'Aviation Civile (ENAC) in 1992. He has been a design engineer at the Centre d'Etudes 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 currenlty the deputy head of the DSNA/R&D department.