AIR TRAFFIC CONTROL WITH AN EXPECTED VALUE CRITERION [1]

Andrea Lecchini * **William Glover** * **John Lygeros** ** **Jan Maciejowski** *

* *Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK,*
{al394, wg214, jmm}@eng.cam.ac.uk
** *Department of Electrical and Computer Engineering, University of Patras, Rio, Patras, GR-26500, GREECE,*
lygeros@ee.upatras.gr

Abstract: In this contribution we discuss a stochastic framework for air traffic conflict resolution. The conflict resolution task is posed as the problem of optimizing an expected value criterion. Optimization is carried out by Monte Carlo Markov Chain (MCMC) simulation. A numerical example illustrates the proposed strategy.

Keywords: Air Traffic Control, Monte Carlo Markov Chain, Model Predictive Control

1. INTRODUCTION

In the current organisation of Air Traffic Management the centralised Air Traffic Control (ATC) is in complete control of the air traffic and ultimately responsible for safety. Before take off, aircraft receive flight plans which cover the entire flight. During the flight, ATC sends additional instructions to them, depending on the actual traffic, in order to improve traffic flow and avoid dangerous encounters. The main objective of ATC is to maintain safe separation. The level of accepted minimum safe separation can vary with the density of the traffic and the region of airspace. For example, a largely accepted value for horizontal minimum safe separation is 5 nmi in general en-route airspace which is reduced to 3 nmi in approach sectors for aircraft landing and departing. A conflict is defined as loss of minimum safe separation between two aircraft. If it is possible, ATC tries also to fulfil the, possibly conflicting, requests of aircraft and airlines (desired path to avoid turbulence, desired time of arrivals to meet schedule, etc..).

To improve performance of ATC, mainly in view of increasing levels of traffic, research effort has been devoted over the last decade to create tools for Conflict Detection and Conflict Resolution. For a review of this aspect of ATC research see (Kuchar and Yang, 2000). In Conflict Detection one has to evaluate the possibility of future conflict starting from the current state of the airspace and taking into account uncertainty in the future position of aircraft while they follow their flight plans. For this task one needs a model to predict the future. In a probabilistic setting, the model could be either an empirical distribution of future position or a stochastic differential equation that describes the aircraft motion and defines implicitly a distribution for future aircraft positions. The stochastic part enters the system as the action of the wind field and several uncertainties in the physics of the aircraft. On the basis of the prediction model one can evaluate metrics related to safety. One example of a possible metric is conflict probability over a certain time horizon. Several methods have been developed to

[1] Work supported by the European Commission under project HYBRIDGE, IST-2001-32460 and by Eurocontrol under project C20051E/BM/03.

estimate different metrics related to safety for a number of prediction models, e.g (Blom and Bakker, 2002; Hu *et al.*, 2003; Irvine, 2001; Paielli and Erzberger, 1997; Prandini *et al.*, 2000).

Among other methods, Monte Carlo (MC) methods have the main advantage of allowing flexibility in the complexity of the prediction model since the model is used only as a simulator and, in principle, it is not involved in explicit calculations. In all methods a trade off exists between computational effort (simulation time in the case of MC methods) and accuracy of the model. Techniques to accelerate MC methods especially for rare event computations are under development, see e.g. (Krystul and Blom, 2004).

For Conflict Resolution one wants to calculate suitable maneuvers to avoid a predicted conflict. A number of Conflict Resolution algorithms has been proposed for a deterministic setting, see (Kuchar and Yang, 2000). In a stochastic setting, the research effort has been concentrated mainly on Conflict Detection, with only a few resolution strategies proposed. Simple conflict resolution maneuvers in a stochastic setting have been considered in (Paielli and Erzberger, 1997; Prandini *et al.*, 2000).

In this paper we present a Monte Carlo Markov Chain (MCMC) framework (Robert and Casella, 1999) for Conflict Resolution in a stochastic setting. The approach is borrowed from Bayesian statistics (Mueller, 1999; Mueller *et al.*, 2002). We will consider a resolution criterion that takes into account separation and other factors (e.g. aircraft requests). Then, the procedure of (Mueller, 1999) is employed to estimate the resolution maneuver that optimises the expected value criterion through MCMC simulation. The interesting point in this approach is that it extends the advantages of Monte Carlo techniques, in terms of flexibility and complexity of the problems that can be tackled, to Conflict Resolution.

The paper is organised as follows. In the next section we recall previous work on the modelling of aircraft motion from the point of view of ATC. In Section 3 we present our approach to conflict resolution based on Monte Carlo optimization techniques. A simulation example is illustrated in Section 4. Conclusions, in Section 5, complete the paper.

2. MODELLING OF AIRCRAFT MOTION

In earlier work we developed an air-traffic simulator that simulates adequately the behaviour of a set of aircraft from the point of view of ATC. The simulator implements realistic models of current commercial aircraft described in the Base of Aircraft Data (BADA) (EUROCONTROL Experimental Centre, 2002). The simulator contains also realistic stochastic models of the wind disturbance (Cole *et al.*, 1998). The aircraft models contain continuous dynamics, arising from the physical motion of the aircraft, discrete dynamics, arising from the logic embedded in the Flight Management System, and stochastic dynamics, arising from the effect of the wind and incomplete knowledge of physical parameters (for example, the aircraft mass, which depends on fuel, cargo and number of passengers). The simulator has been coded in Java and can be used in different operation modes either to generate accurate data, for validation of the performance of conflict detection and resolution algorithm, or to run faster simulations of simplified models. The nominal path for each aircraft is entered in the simulator as a sequence of way-points. The actual trajectories of the aircraft are then a perturbed version of the nominal path, depending on the particular realizations of wind disturbances and uncertain parameters. The reader is referred to (Glover and Lygeros, 2003) for a more detailed description of the simulator.

3. MONTE CARLO OPTIMIZATION OF AN EXPECTED VALUE CRITERION

In our approach we formulate conflict resolution as a constrained optimization problem. Given a set of aircraft involved in a conflict, the conflict resolution maneuver is determined by a parameter ω (e.g. a sequence of way points or vector commands) which defines the nominal paths of the aircraft. The actual execution of the maneuver is affected by uncertainty. Therefore, the sequence of actual positions of the aircraft (for example: the sequence of positions every 6 seconds which is a typical time interval between two successive radar sweeps) during the resolution maneuver is, *a-priori* of its execution, a random variable denoted by X. A conflict is defined as the event that the positions of two aircraft during the execution of the maneuver get too close. The objective is to select ω in order to maximise the expected value of some measure of performance associated to the execution of the resolution maneuver while ensuring a small probability of conflict. In this section we introduce the formulation of the problem in a general fashion.

3.1 Penalty formulation of an expected value optimization problem with constraints

Let X be a random variable whose distribution depends on some parameter ω. The distribution of X is denoted by $p_\omega(x)$ with $x \in \mathbf{X}$. The set of all possible values of ω is denoted by $\mathbf{\Omega}$. We assume that a constraint on the random variable X is given in terms of a feasible set $\mathbf{X_f} \subseteq \mathbf{X}$. We say that a realization x, of random variable X, violates the constraint if $x \notin \mathbf{X_f}$. Moreover, we assume that for a realization $x \in \mathbf{X_f}$ some definition of performance of x is given. In general performance can depend also on the value of ω, therefore performance is measured by a function $\mathrm{perf}(\omega, x)$, $x \in \mathbf{X_f}$, $\omega \in \mathbf{\Omega}$. We assume that $\mathrm{perf}(\omega, x)$ takes values in $(0, 1]$. The probability of satisfying the constraint is denoted by $\mathrm{P}(\omega)$

$$\mathrm{P}(\omega) = \int_{x \in \mathbf{X_f}} p_\omega(x) dx \,. \tag{1}$$

The probability of violating the constraint is denoted by $\bar{\mathrm{P}}(\omega) = 1 - \mathrm{P}(\omega)$. The expected performance for a given $\omega \in \mathbf{\Omega}$ is denoted by $\textsc{Perf}(\omega)$, where

$$\textsc{Perf}(\omega) = \int_{x \in \mathbf{X_f}} \mathrm{perf}(\omega, x) p_\omega(x) dx \,. \tag{2}$$

Ideally one would like to maximise the performance over all ω, subject to a bound on the probability of constraint satisfaction. Given a bound $\bar{\mathbf{P}} \in [0, 1]$, this corresponds to solving the constrained optimization problem

$$\textsc{Perf}_{\max|\bar{\mathbf{P}}} = \sup_{\omega \in \mathbf{\Omega}} \textsc{Perf}(\omega) \tag{3}$$
$$\text{subject to } \bar{\mathrm{P}}(\omega) < \bar{\mathbf{P}}. \tag{4}$$

Clearly, a necessary condition for the problem to have a solution is that there exists $\omega \in \mathbf{\Omega}$ such that $\bar{\mathrm{P}}(\omega) < \bar{\mathbf{P}}$, or, equivalently,

$$\bar{\mathrm{P}}_{\min} = \inf_{\omega \in \mathbf{\Omega}} \bar{\mathrm{P}}(\omega) < \bar{\mathbf{P}}. \tag{5}$$

This optimization problem is generally difficult to solve, or even to approximate by randomised methods. Here we approximate this problem by an optimization problem with penalty terms. We show that with a proper choice of the penalty term we can enforce the desired maximum bound on the probability of violating the constraint, provided that such a bound is feasible, at the price of sub-optimality in the resulting expected performance. Let us introduce the function $u(\omega, x)$ defined as

$$u(\omega, x) = \begin{cases} \mathrm{perf}(\omega, x) + \Lambda & x \in \mathbf{X_f} \\ \\ 1 & x \notin \mathbf{X_f}, \end{cases} \tag{6}$$

where $\Lambda > 1$. The parameter Λ represents a reward for constraint satisfaction. The expected value of $u(\omega, x)$ is given by

$$U(\omega) = \int_{x \in \mathbf{X}} u(\omega, x) p_\omega(x) dx \qquad \omega \in \mathbf{\Omega} \,. \tag{7}$$

Instead of the constrained optimization problem (3)–(4) we solve the unconstrained optimization problem:

$$U_{\max} = \sup_{\omega \in \mathbf{\Omega}} U(\omega). \tag{8}$$

Assume the supremum is attained and let $\bar{\omega}$ denote the optimum solution, i.e. $U_{\max} = U(\bar{\omega})$. The following proposition introduces bounds on the probability of violating the constraints and the level of sub-optimality of $\textsc{Perf}(\bar{\omega})$ over $\textsc{Perf}_{\max|\bar{\mathbf{P}}}$.

Proposition 3.1. The maximiser, $\bar{\omega}$, of $U(\omega)$ satisfies

$$\bar{\mathrm{P}}(\bar{\omega}) \leq \frac{1}{\Lambda} + \frac{\Lambda - 1}{\Lambda} \bar{\mathrm{P}}_{\min}$$
$$\textsc{Perf}(\bar{\omega}) \geq \textsc{Perf}_{\max|\bar{\mathbf{P}}} - (\Lambda - 1)(\bar{\mathbf{P}} - \bar{\mathrm{P}}_{\min})$$

Proof: see (Lecchini *et al.*, 2005).

Proposition 3.1 suggests a method for choosing Λ to ensure that the solution $\bar{\omega}$ of the optimization problem will satisfy $\bar{\mathrm{P}}(\bar{\omega}) \leq \bar{\mathbf{P}}$. In particular it suffices to know $\bar{\mathrm{P}}(\omega)$ for some $\omega \in \mathbf{\Omega}$ with $\bar{\mathrm{P}}(\omega) < \bar{\mathrm{P}}$ to obtain a bound. If there exists $\omega \in \mathbf{\Omega}$ for which $\hat{\mathrm{P}} = \bar{\mathrm{P}}(\omega)$ is known, then any

$$\Lambda \geq \frac{1 - \hat{\mathrm{P}}}{\bar{\mathbf{P}} - \hat{\mathrm{P}}} \tag{9}$$

ensures that $\bar{\mathrm{P}}(\bar{\omega}) \leq \bar{\mathbf{P}}$. If we know that there exists a parameter $\omega \in \mathbf{\Omega}$ for which the constraints are satisfied almost surely, a tighter (and potentially more useful) bound can be obtained:

$$\Lambda \geq \frac{1}{\bar{\mathbf{P}}} \tag{10}$$

ensures that $\bar{\mathrm{P}}(\bar{\omega}) \leq \bar{\mathbf{P}}$. Clearly to minimise the gap between the optimal performance and the performance of $\bar{\omega}$ we need to select Λ as small as possible. Therefore the optimal choices of Λ that ensure the bounds on constraint satisfaction and minimise the sub-optimality of the solution are $\Lambda = \frac{1-\hat{\mathrm{P}}}{\bar{\mathbf{P}}-\hat{\mathrm{P}}}$ and $\Lambda = \frac{1}{\bar{\mathbf{P}}}$ respectively.

3.2 Simulation-based optimization

In this subsection we recall a simulation-based procedure, to find approximate optimisers of $U(\omega)$. The only requirement for applicability of the procedure is to be able to obtain realizations of the random variable X with distribution $p_\omega(x)$ and to evaluate $u(\omega, x)$ point-wise. This optimization procedure is in fact a general procedure for the optimization of expected value criteria. It has been originally proposed in the Bayesian statistics literature (Mueller, 1999).

The optimization strategy relies on extractions of a random variable Ω whose distribution has modes which coincide with the optimal points of $U(\omega)$. These extractions are obtained through Monte Carlo Markov Chain (MCMC) simulation (Robert and Casella, 1999). The problem of optimizing the expected criterion is then reformulated as the problem of estimating the optimal points from extractions concentrated around them. In the optimization procedure, there exists a tunable trade-off between estimation accuracy of the optimiser and computational effort. In particular, the distribution of Ω is proportional to $U(\omega)^J$ where J is a positive integer which allows the user to increase the "peakedness" of the distribution and concentrate the extractions around the modes at the price of an increased computational load. If the tunable parameter J is increased during the optimization procedure, this approach can be seen as the counterpart of Simulated Annealing for a stochastic setting. Simulated Annealing is a randomised optimization strategy developed to find tractable approximate solutions to complex deterministic combinatorial optimization problems. A formal parallel between these two strategies has been derived in (Mueller *et al.*, 2002).

The MCMC optimization procedure can be described as follows. Consider a stochastic model formed by a random variable Ω, whose distribution has not been defined yet, and J conditionally independent replicas of random variable X with distribution $p_\Omega(x)$. Let us denote $h(\omega, x_1, x_2, \ldots, x_J)$ the joint distribution of $(\Omega, X_1, X_2, X_3, \ldots, X_J)$. It is straightforward to see that if

$$h(\omega, x_1, x_2, \ldots, x_J) \propto \prod_j u(\omega, x_j) p_\omega(x_j) \quad (11)$$

then the marginal distribution of Ω, also denoted by $h(\omega)$ for simplicity, satisfies

$$h(\omega) \propto \left[\int u(\omega, x) p_\omega(x) dx \right]^J = U(\omega)^J . \quad (12)$$

This means that if we can extract realizations of $(\Omega, X_1, X_2, X_3, \ldots, X_J)$ then the extracted Ω's will be concentrated around the optimal points of $U(\Omega)$ for a sufficiently high J. These extractions can be used to find an approximate solution to the optimization of $U(\omega)$.

Realizations of the random variables $(\Omega, X_1, X_2, X_3, \ldots, X_J)$, with the desired joint probability density given by (11), can be obtained through Monte Carlo Markov Chain simulation. The algorithm is presented below. In the algorithm, $g(\omega)$ is known as the *instrumental* (or *proposal*) distribution and is freely chosen by the user; the only requirement is that $g(\omega)$ covers the support of $h(\omega)$.

MCMC ALGORITHM (METROPOLIS-HASTINGS)

Initial state $(\bar{\omega}, \bar{x}_j \, j = 1, \ldots, J)$ and $\bar{u}_J = \prod_j u(\bar{\omega}, \bar{x}_j)$

1 Extract

$$\tilde{\Omega} \sim g(\omega|\bar{\omega})$$

2 Extract

$$\tilde{X}_j \sim p_{\tilde{\Omega}}(x) \quad j = 1 \ldots J$$

and calculate

$$\tilde{U}_J = \prod_j u(\tilde{\Omega}, \tilde{X}_j)$$

3 Extract the new state of the chain as

$$(\bar{\Omega}, \bar{U}_J) = \begin{cases} (\tilde{\Omega}, \tilde{U}_J) & \text{with prob. } \rho(\bar{\omega}, \bar{u}_J, \tilde{\Omega}, \tilde{U}_J) \\ (\bar{\omega}, \bar{u}_J) & \text{with prob. } 1 - \rho(\bar{\omega}, \bar{u}_J, \tilde{\Omega}, \tilde{U}_J) \end{cases}$$

where

$$\rho(\bar{\omega}, \bar{u}_J, \tilde{\omega}, \tilde{u}_J) = \min \left\{ 1, \frac{\tilde{u}_J}{\bar{u}_J} \frac{g(\bar{\omega}|\tilde{\omega})}{g(\tilde{\omega}|\bar{\omega})} \right\}$$

4 Repeat steps 1 through 3

This algorithm is a formulation of the Metropolis-Hasting algorithm for a desired distribution given by $h(\omega, x_1, x_2, \ldots, x_J)$ and proposal distribution given by

$$g(\omega) \prod_j p_\omega(x_j) .$$

In this case, the acceptance probability for the standard Metropolis-Hastings algorithm is

$$\frac{h(\tilde{\omega}, \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_J)}{h(\omega, x_1, x_2, \ldots, x_J)} \frac{g(\omega) \prod_j p_\omega(x_j)}{g(\tilde{\omega}) \prod_j p_\omega(\tilde{x}_j)} .$$

By inserting (11) in the above expression one obtains the probability $\rho(\omega, u_J, \tilde{\omega}, \tilde{u}_J)$. Under minimal assumptions, the Markov Chain $\Omega(k)$ is uniformly ergodic with stationary distribution $h(\omega)$ given by (12). Results that characterize the convergence rate to the stationary distribution can be found for example in (Robert and Casella, 1999). A general guideline to obtain faster convergence is to concentrate the search distribution $g(\omega)$ where $U(\omega)$ assumes nearly optimal values.

4. SIMULATION EXAMPLE

We consider the problem of sequencing two aircraft. This is typically a task of air-traffic controllers in Terminal Airspace where aircraft descend from cruising altitude and need to be sequenced and separated by a certain time interval before entering in the Final Approach Sector. In Figure 1 several possible trajectory realizations of a descending aircraft corresponding to the same nominal path are displayed. In this figure, the aircraft descends from 35000 *ft* to 10000 *ft*. In addition to stochastic wind terms, uncertainty about the mass of the aircraft is introduced as a uniform distribution between two extreme values. The figure suggests that the resulting uncertainty in the position of aircraft is of the order of magnitude of some kilometres.

We consider the problem of sequencing two descending aircraft as illustrated in Figure 2-(a). The initial position of the first aircraft (A1) is $[-100000 \; 100000]$ (where coordinates are expressed in meters) and altitude 35000 *ft*. The path of this aircraft is fixed. This aircraft proceeds to way-point $[-90000 \; 90000]$ where it will start a descent to 15000 *ft*. The trajectory of A1, while descending, is determined by an intermediate way-point in $[0 \; 0]$ and a final way-point in $[100000 \; 0]$, where aircraft exit the sector. The second aircraft (A2) is initially in $[-100000 \; -100000]$ and altitude 35000 *ft*. This aircraft proceeds to way-point $[-90000 \; -90000]$ where it will start

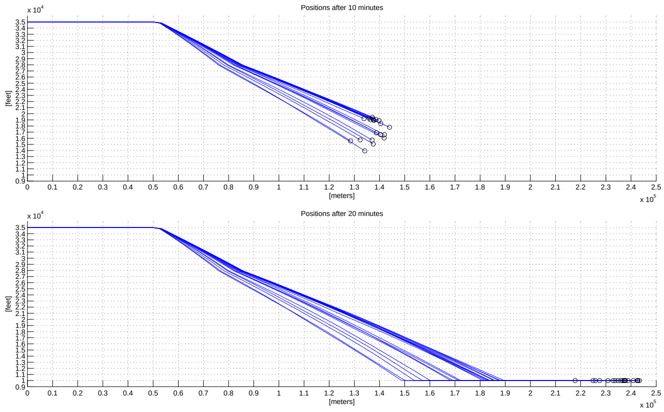
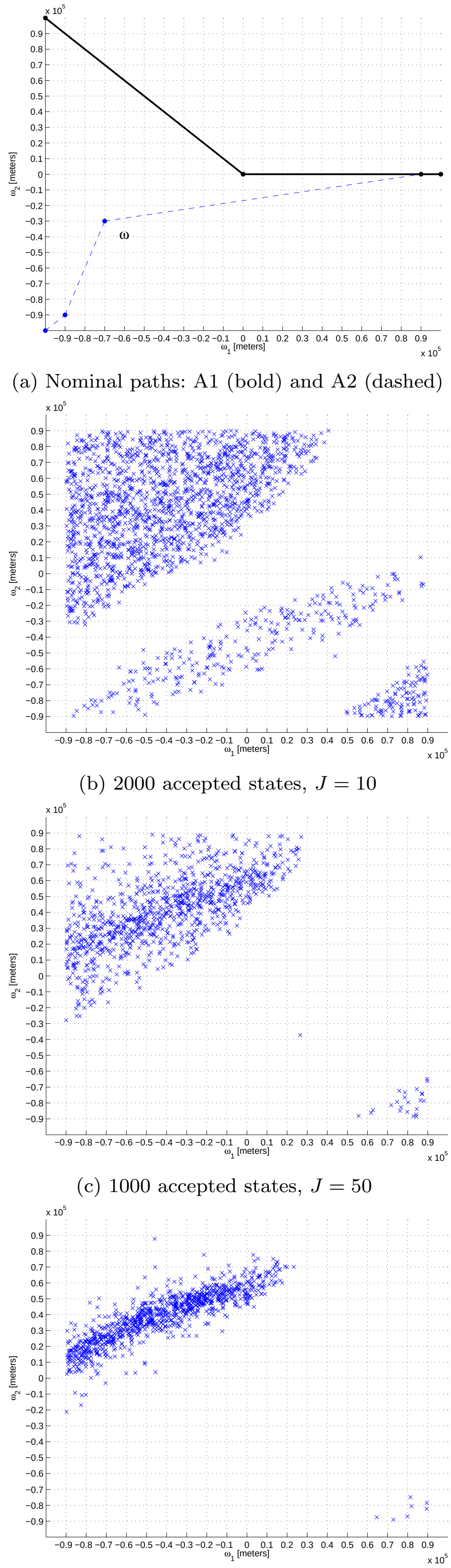


Fig. 1. Trajectory realizations of aircraft descent

the descent to 15000 *ft*. The intermediate way-point $\omega = [\omega_1 \; \omega_2]$ must be selected in the range $\omega_1, \omega_2 \in [-90000 \; 90000]$. The aircraft will then proceed to way-point [90000 0] and then to the final way-point [100000 0].

We assume that the objective is to obtain a time separation of 300 *sec* between the arrivals of the two aircraft at the final way-point. Performance in this sense is measured by $\text{perf} = e^{-a\cdot(|T_1-T_2|-300)}$ where T_1 and T_2 are the arrival times of the aircraft at the final way-point and $a = 5 \cdot 10^{-3}$. The constraint is that the trajectory of the two aircraft are not conflicting. A conflict is defined as the situation of loss of minimum safe separation. Safe separation is defined by a protected zone centred around each aircraft having radius 5 *nmi* and height 2000 *ft*, so that aircraft which do not have 5 *nmi* of horizontal separation must have 1000 *ft* of vertical separation. We optimise initially with an upper bound on probability of constraint violation given by $\bar{\mathbf{P}} = 0.1$. It is easy to see that there exists a maneuver in the set of optimization parameters that gives negligible conflict probability. Therefore, based on inequality (10), we select $\Lambda = 10$ in the optimization criterion.

The results of the optimization procedure are illustrated in Figures 2-(b-d). Each figure shows the scatter plot of the accepted parameters during MCMC simulation for different choices of J and search distribution g. In all cases the first 10% of accepted parameters was discarded as a *burn in* period to allow convergence of the chain to its stationary distribution. For each case we give also the ratio between accepted and proposed states during MCMC simulation. Figure 2.(b) illustrates the case $J = 10$. In this case the proposal distribution g was uniform over the parameter space. The ratio accepted/proposed states was 0.27. Regions characterised by a low density of accepted parameters can be clearly seen in the figure. These are parameters which correspond to nominal paths with high probability of conflict. The figure also shows distinct "clouds" of accepted maneuvers. They correspond to different sequences of arrivals:



(a) Nominal paths: A1 (bold) and A2 (dashed)

(b) 2000 accepted states, $J = 10$

(c) 1000 accepted states, $J = 50$

(d) 1000 accepted states, $J = 100$

Fig. 2. Accepted states during MCMC simulation

either A1 arrives before before A2 (top left and bottom right clouds) or A1 arrives after A2 (middle cloud). Figure 2.(c) illustrates the case $J = 50$. In this case the proposal distribution g was a sum of 2000 Gaussian distributions $N(\mu, \sigma^2 I)$ with variance $\sigma^2 = 10^7\, m^2$. The means of Gaussian distributions were 2000 parameters randomly chosen from those accepted in the MCMC simulation for $J = 10$. The choice of this proposal distribution gives clear computational advantages since less computational time is spent searching over regions of non optimal parameters. In this case the ratio accepted/proposed states was 0.34. Figure 2.(d) illustrates the case $J = 100$ and proposal distribution constructed as before from states accepted for $J = 50$. Here the ratio accepted/proposed states was 0.3. Figure 2.(d) indicates that a nearly optimal maneuver is $\omega_1 = -40000$ and $\omega_2 = 40000$. The probability of conflict for this maneuver, estimated by 1000 Monte-Carlo runs, was zero. The estimated expected time separation between arrivals was 283 *sec*.

5. CONCLUSIONS

In this paper we illustrated our current approach to air traffic conflict resolution in a stochastic setting based on the use of Monte Carlo methods. The main motivation for our approach is to enable the use of realistic stochastic hybrid models of aircraft flight; Monte Carlo methods appear to be the only ones that allow such models. We have formulated conflict resolution as the optimization of an expected value criterion with probabilistic constraints. Here, a penalty formulation of the problem has been considered which guarantees constraint satisfaction but delivers a suboptimal solution. A side effect of the optimization procedure is that structural differences between maneuvers are highlighted as "clouds" of maneuvers accepted by the algorithm.

Our current research is concerned with overcoming the sub optimality imposed by the need to provide constraint satisfaction guarantees. A possible way is to use the Monte Carlo Markov Chain procedure presented in Section 3 to obtain optimization parameters that satisfy the constraint and then to optimise over this set in a successive step. Formulation of the conflict resolution procedure in the Sequential Monte Carlo framework (Doucet *et al.*, 2001) is also under investigation. Finally, we are also working on modelling and implementation in the simulator of typical Air-Traffic Control situation with a realistic parameterisation of control actions and control objectives.

REFERENCES

Blom, H.A.P. and G.J. Bakker (2002). Conflict probability and incrossing probability in air traffic management. In: *IEEE Conference on Decision and Control*. Las Vegas, Nevada, U.S.A.

Cole, R.E., C. Richard, S. Kim and D. Bailey (1998). An assessment of the 60 km rapid update cycle (ruc) with near real-time aircraft reports. Technical Report NASA/A-1. MIT Lincoln Laboratory.

Doucet, A., N de Freitas and N. Gordon (eds) (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.

EUROCONTROL Experimental Centre (2002). *User Manual for the Base of Aircraft Data (BADA)*. Available from World Wide Web: http://www.eurocontrol.fr/projects/bada/.

Glover, W. and J. Lygeros (2003). A multi-aircraft model for conflict detection and resolution algorithm validation. Technical Report WP1, Deliverable D1.3. HYBRIDGE. Available from World Wide Web: http://www.nlr.nl/public/hosted-sites/hybridge/.

Hu, J., M. Prandini and S. Sastry (2003). Aircraft conflict detection in presence of spatially correlated wind perturbations. In: *AIAA Guidance, Navigation and Control Conf.*. Austin, Texas, USA.

Irvine, R. (2001). A geometrical approach to conflict probability estimation. In: *4th USA/Europe Air Traffic Management R&D Seminar*. Budapest, Hungary.

Krystul, J. and H.A.P. Blom (2004). Monte Carlo simulation of rare events in hybrid systems. Technical Report WP8, Deliverable D8.3. HYBRIDGE. Available from World Wide Web: http://www.nlr.nl/public/hosted-sites/hybridge/.

Kuchar, J.K. and L.C. Yang (2000). A review of conflict detection and resolution methods. *IEEE Transactions on Intelligent Transportation Systems* **1**(4), 179–189.

Lecchini, A., W. Glover, J. Lygeros and J. Maciejowski (2005). Air-Traffic Control in Approach Sectors: simulation examples and optimisation. In: *HSCC: Hybrid Systems Computation and Control*. Springer Lecture Notes in Computer Science. Zurich.

Mueller, P. (1999). Simulation based optimal design. In: *Bayesian Statistics 6*. J.O. Berger, J.M. Bernardo, A.P. Dawid and A.F.M. Smith (eds.), Oxford University Press. pp. 459–474.

Mueller, P., B. Sanso and M. De Iorio (2002). Optimal bayesian design by inhomogeneous markov chain simulation. *Submitted*.

Paielli, R.A. and H. Erzberger (1997). Conflict probability estimation for free flight. *Journal of Guidance, Control and Dynamics* **20**(3), 588–596.

Prandini, M., J. Hu, J. Lygeros and S. Sastry (2000). A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation Systems*.

Robert, C.P. and G. Casella (1999). *Monte Carlo Statistical Methods*. Springer-Verlag.